7-1-2007

VTAC: Virtual terrain assisted impact assessment for cyber attacks

Brian Argauer

VTAC: Virtual Terrain Assisted Impact Assessment for Cyber Attacks

by

Brian John Argauer Jr.

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

Supervised by

Assistant Professor, Department of Computer Engineering Dr. Shanchieh Jay Yang
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
July 2007

Approved By:

Dr. Shanchieh Jay Yang
Assistant Professor, Department of Computer Engineering
Primary Adviser

Professor Warren R. Carithers
Associate Professor, Department of Computer Science

Dr. Fei Hu
Assistant Professor, Department of Computer Engineering

Thesis Release Permission Form

Rochester Institute of Technology

Kate Gleason College of Engineering

Title:

I, Brian John Argauer Jr., hereby grant permission to the Wallace Memorial Library reporduce my thesis in whole or part.

Brian John Argauer Jr.

Date

Dedication

To my family and friends who have provided support and encouragement throughout my life. Also, to my fiancée, who has loved and supported me through the course of this thesis and all of my collegiate endeavors. Finally, to my siblings and parents, for their unconditional love, guidance, and motivation.

Acknowledgments

First and foremost, I would like to thank Dr. Yang for being a very accommodating and supportive adviser who has provided valuable guidance toward the completion of this thesis. I am also appreciative for his discussions and contribution to this work. I would also like to thank Professor Carithers and Dr. Hu for serving on my committee.

I would like to thank Jared Holsopple of CUBRC for providing valuable input to the design of VTAC. A special thanks to fellow classmates from the RIT Computer Engineering Department, especially Stephen Byers, Gilbert Hendry, and Eric Ernst for their support and sharing their research knowledge. Finally, to Jon Claudius and Stephen Byers for their expert insight into network configurations.

This work is funded through the National Center for Multisource Information Fusion (NCMIF) grant under the technical supervision of AFRL/IFEA.

Abstract

Recently, there has been substantial research in the area of network security. Correlation of intrusion detection sensor alerts, vulnerability analysis, and threat projection are all being studied in hopes to relieve the workload that analysts have in monitoring their networks. Having an automated algorithm that can estimate the impact of cyber attacks on a network is another facet network analysts could use in defending their networks and gaining better overall situational awareness.

Impact assessment involves determining the effect of a cyber attack on a network. Impact algorithms may consider items such as machine importance, connectivity, user accounts, known attacker capability, and similar machine configurations. Due to the increasing number of attacks, constantly changing vulnerabilities, and unknown attacker behavior, automating impact assessment is a non-trivial task. This work develops a virtual terrain that contains network and machine characteristics relevant to impact assessment. Once populated, this virtual terrain is used to perform impact assessment algorithms.

The goal of this work is to investigate and propose an impact assessment system to assist network analysts in prioritizing attacks and analyzing overall network status. VTAC is tested with several scenarios over a network with a variety of configurations. Insights into the results of the scenarios, including how the network topologies and network asset configurations affect the impact analysis are discussed.

Contents

List of Figures

List of Tables

Glossary

D

DC Domain controller, p. 59.

DDoS Distibuted Denial of Service attack, p. 7.

DMZ Demilitarized zone, p. 46.

I

IDS Intrusion detection system, p. 1.

I_H Impact score for a host, p. 31.

I_N Impact score for the network, p. 31.

I_S Impact score for a service, p. 31.

I_U Impact score for a user, p. 31.

I_X Impact score for the component X; either host, service, user, or network, p. 36.

J

JDL Joint Director of Laboratories, p. 3.

M

ME_H Impact score with Maximum Exposure for all Hosts, p. 36.

ME_S Impact score with Maximum Exposure for asserted Services, p. 36.

V

VTAC Virtual Terrain assisted impact Assessment for Cyber attacks, p. 1.

Chapter 1

Introduction

This work develops a system to perform impact assessment of a network and its components based on detected cyber attacks. VTAC: Virtual Terrain assisted impact Assessment for Cyber attacks, is the framework created to perform impact assessment. VTAC contains a virtual terrain model used to store and relate the necessary ingredients needed to perform a network impact assessment. Additionally, it incorporates algorithms to process the terrain and incoming attack actions so as to deduce the impact of cyber attacks.

1.1 Motivation

A modern day network has Intrusion Detection Systems (IDSs) that provide network analysts with alerts of malicious actions. Analysts typically examine these alerts manually and are more than often overwhelmed by alert volumes, allowing attacks to slip through [1]. With 4882 vulnerabilities reported in 2005 and 6604 in 2006, approximately a 35% increase [19], the number of vulnerabilities are simply becoming too large for analysts to keep track of and understand. Out of the 6604 vulnerabilities from 2006, 85% of them can be attacked remotely, and 65% lead to disruption of service [19]. An FBI/CSI survey showed that in 2002, 80% of respondents indicated a financial loss as a result of a computer breach, and 25% reported intrusions to law enforcement [6]. These statistics help show the need of a better line of defense for network intrusions.

The task of assessing cyber attacks has drawn increasing attention from the information

fusion community. Drawing analogies from traditional fusion problems, assessing cyber attacks involves detecting, tracking, correlating, analyzing the impact of, and projecting attack movements. Malicious activity on computer networks trigger IDSs to produce alerts. Each attack action may trigger zero, one, or many alert messages. Correlating and filtering alert messages, i.e., observables, provide traces or tracks of ongoing multistage attacks in a computer network. Precise and timely impact analysis and predictions shall lead to better decision making and minimal operational interruptions when combating cyber attacks. The focus of this work is on assessing the impact of the tracked cyber attacks.

Assessing the impact of cyber attack actions depends on the detection and tracking of malicious activity. Host-based and network based IDSs typically monitor application and operating system level activity as well as network traffic. Alerts are generated when monitored activity matches one or more signatures of previously known attacks (signature-based) or is abnormal and suspicious (anomaly-based). Non-intrusive and intrusive malicious activity detection has been widely tackled yet still continuously poses challenges, due to the constantly evolving nature of vulnerabilities and, consequently, changes in exploitation mechanisms. As a result, alert messages produced by IDSs may be incomplete and misleading.

In 2000, Bass [1] advocated the need of information fusion when facing overwhelming number of alerts reported on typical enterprise networks. Since then, much work has been devoted to alert correlation, e.g., INFERD [26], [27], TIAA [20], and MIRADOR [3]. Correlating IDS alerts involves reasoning based on, primarily, the source and target IP addresses, the attack type descriptions from the alert messages, and the time interval between alerts. This set of information helps revealing the courses of action potentially taken by multistage attacks, which may span over multiple machines or subnets. Alerts that belong to the same course of action are grouped and traced to form an attack track. Each attack track, which may be modeled as a directed graph, illustrates the causal and sequential relationships between alerts belonging to the same multistage attack. Note that undetected activity and excessive alerts (typically due to reconnaissance activity) may lead

to mis-correlated alerts or fragmented attack tracks.

1.2 What is Impact Assessment?

While alert correlation is still under investigation for better accuracy and real-time operation, the next challenge is to infer future attack actions and assess the impact of attack actions based on the detected attack tracks. Threat and impact assessment of cyber attacks may be categorized as a L3 fusion problem based on the Joint Directors of Laboratories (JDL) fusion model [4], [7] and its revision in 2004 [15]. Impact assessment determines the consequences that attack actions have on a network and its assets. Threat assessment involves determining the capability, opportunity, and intent of attack actions, *e.g.*, TANDI [8]. The JDL fusion model is built from knowledge and information from the ground up with goals of high-level situational and environmental awareness in mind. Traditionally, these techniques have been applied to the location, characterization, and identification of dynamic entities such as emitters, platforms, and weapons that would be of concern to the military [7]. At the lowest level, the existence of an entity is acknowledged, and by applying estimation techniques, such as Kalman filter, to the timeseries of inherently noisy measurements, position and velocity may be calculated. At higher levels, Bayesian Analysis or Dempster-Shafer combination may be used to establish the identity or confidence of an entity measurement. Rule-based reasoning systems may provide situation assessment and threat analysis inferring the intent of the enemy.

Figure 1.1 shows a mapping of the JDL fusion levels to cyber security problems. This mapping divides the problem space into modular sub-tasks. Given the track estimates from Level 1, this work focuses on estimating the severity of current attack steps on the network and its assets. Information about network configuration, software vulnerabilities, and the criticality or importance of each asset is the basis to provide accurate impact assessment. Striped modules in Figure 1.1 are ones that will be worked on in this thesis.

Typically, impact is thought of as the result or consequence that some action has on a

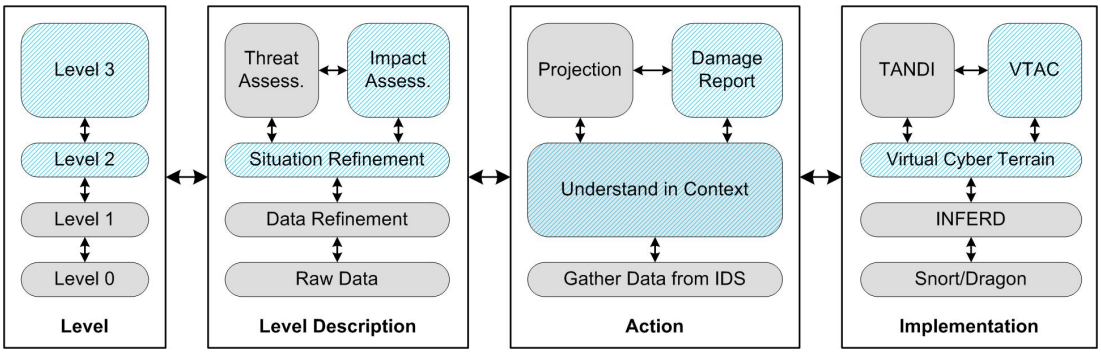

Figure 1.1: JDL fusion levels corresponding to network security

particular entity. Within the cyber domain, this can be interpreted as the potential severity that a cyber attack may have on the existing network given the current situation. To better illustrate the concept of impact, an example of a traditional warfare scenario may be used. Consider a missile launch with potential trajectories indicating impact sites ranging from a large population center to rural farmlands. The impact assessment could be performed in terms of several metrics, for example, loss of life to capital or infrastructure damage. Therefore, one may want to invest their limited defense funding into anti-missile systems for the areas where an impact would result in the most detrimental assessment. In terms of actually predicting the assessment, one would like to know, for example, the type of missile that is launched and the characteristics of its target, which would result in a specific type of outcome.

Roy, Paradis, and Allouche [24] discuss a typical warfare example of an unknown entity (red team) coming into range of a specified defending unit (blue team). A number of metrics are taken into account when determining the threat and eventually impact that this entity could have. One has to consider the velocity and angle that the entity is coming towards the defending unit, what type of weapons they may be carrying and at what range they can reach the defending unit with those weapons. They also mention Closest Point of Approach (CPA) and Time-to-CPA (TCPA) as important metrics that are based from the Threat Reference Point (TRP), the position at which the assessment is based [24]. Note that these metrics may correspond to a cyber attack. As a cyber attack happens, it can occur

at different speeds and with different attack paths depending on topology and connectivity. Attack range and capability may depend on the variety of hacking tools and skill that the attacker has available to them. The CPA and TCPA can also be modeled with respect to the topology and connectivity. TRP can be determined by what entities (specific host, entire network, *etc.*) the network analyst would like to monitor.

Intent analysis is another vital piece of impact assessment [24]. However, unless someone is sitting right next to an attacker, it is difficult to find out what their real intent is. Estimating attacker intent is out of the scope of this thesis.

1.3 Related Work

This section summarizes the related work, including what has been done in the area of modeling networks with vulnerabilities, impact assessment, and mapping attacks to consequences.

1.3.1 Network Modeling for Analysis Purposes

Vidalis and Jones [30] proposed the use of a vulnerability tree to identify the types of attacks an attacker could perform to accomplish a goal. Their model requires a separate vulnerability tree for each possible goal, which could be potentially numerous. These trees can identify the most significant vulnerabilities of the system depending on the education level of the hacker.

Philips and Swiler [22] suggested the use of a Bayesian network to model the vulnerabilities for risk analysis. Their model assumes acyclic graphs, which implies that bi-directional connections between hosts must be modeled in separate acyclic graphs.

Massicotte, Couture, Briand, and Labiche [16] discussed ways of introducing contextual information to cross examine with reported IDS alerts and, thus, reduce false positives. Their experiences suggested that contextual information may be derived by utilizing Snort [25], Nessus [10] and Bugtraq [5]. They also suggest that a network be modeled

as objects because network components are inherently modular and each component has sub-components that follow the same behavior. Our model, developed independently of Massicotte's work, shares some similar ideas, yet provides additional network connectivity, user, privilege, and asset criticality information for impact assessment.

1.3.2 Systems with Impact Analysis

Valeur, Vigna, Kruegel, and Kemmerer [29] introduce a comprehensive alert correlation system that contains an impact analysis component. They use the impact analysis component to, "determine the impact of the detected attacks on the operation of the network being monitored and on the assets that are targeted by the malicious activity." Using this information, components following the impact analysis can prioritize attack information. Valeur, *et al.*, use the following information to conduct impact analysis: a service asset database, service heartbeat monitors, service dependencies, and service importance with respect to the overall network. The asset database contains information on each service. When a service is attacked the heartbeat monitor is used to update information about dependent services.

Porras, Fong, and Valdes [23] discuss M-Correlator, a mission-impact-based approach to alert prioritization and aggregation. They mention the problem of merging and analyzing alert information from the growing number of network monitoring devices. Given the topology and mission of a network, M-Correlator's goal is to rank and consolidate incoming alerts based on the degree of threat they pose to the network. They make use of topology vetting, priority mapping, and incident ranking to assess the impact these alerts can have on the network mission and the probability of them being successful. The priority component takes into account the mission of the network by specifying critical incident types and critical computing and data assets, services, and administrative or untrusted user accounts. Porras, *et al.*, use a classification system for alerts, which can be ranked with interest levels specified by an analyst.

1.3.3 Mapping Attack Categories to Consequences

Howard and Longstaff [9] present a taxonomy that provides a timeline and structure of an attack. As seen in Figure 1.2, they state that attackers use specific tools in order to exploit a certain vulnerability. Once the vulnerability is exploited, the attackers use it to perform an action on a specified target in order to attain an unauthorized result, and finally complete their objective. The entire process from start to finish is classified as an incident. Executing a specific action on a target is referred to as the event, and from using the tool until the attacker gets the unauthorized result is specified as the attack.

The list of actions and unauthorized results in Figure 1.2 can potentially be used to categorize, at the highest level, the network attack techniques, and the consequences of a successful attack. The objectives of the overall incident are at a much higher level than

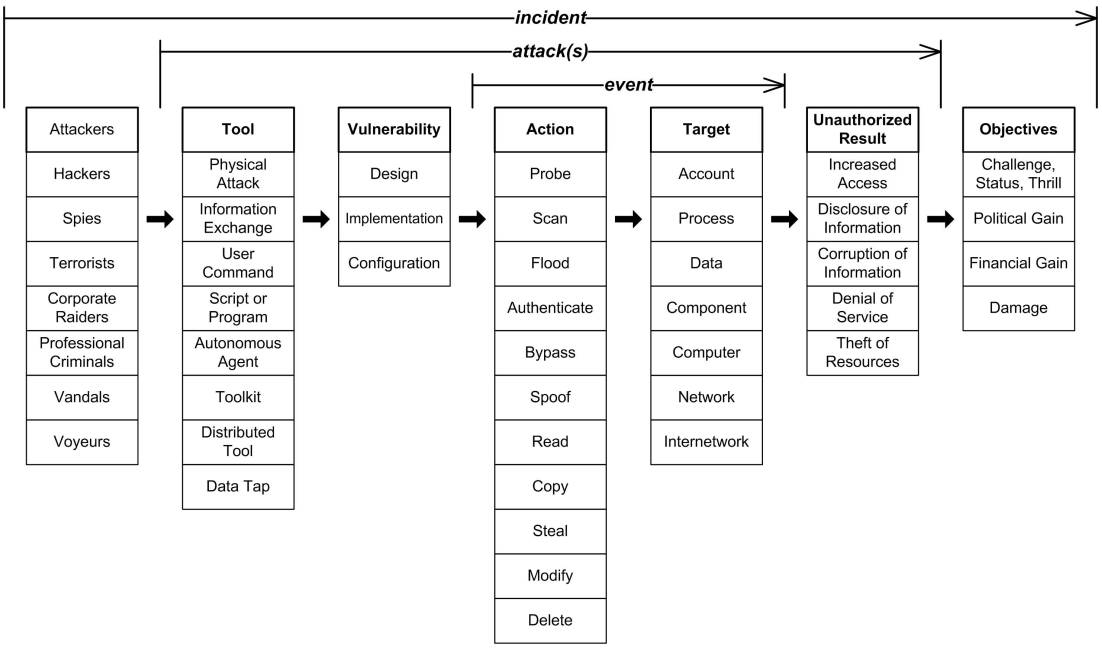

Figure 1.2: Computer and network incident taxonomy [9]

impact assessment will actually assess. Each of the elements in both the attack technique and consequence groups can be divided into smaller partitions based on particular attacks. For example, Mirkovic and Reiher [17] present a taxonomy of Distributed Denial of Service (DDoS) attacks, shown in Figure 1.3. This taxonomy specifies possible ways a DDoS attack

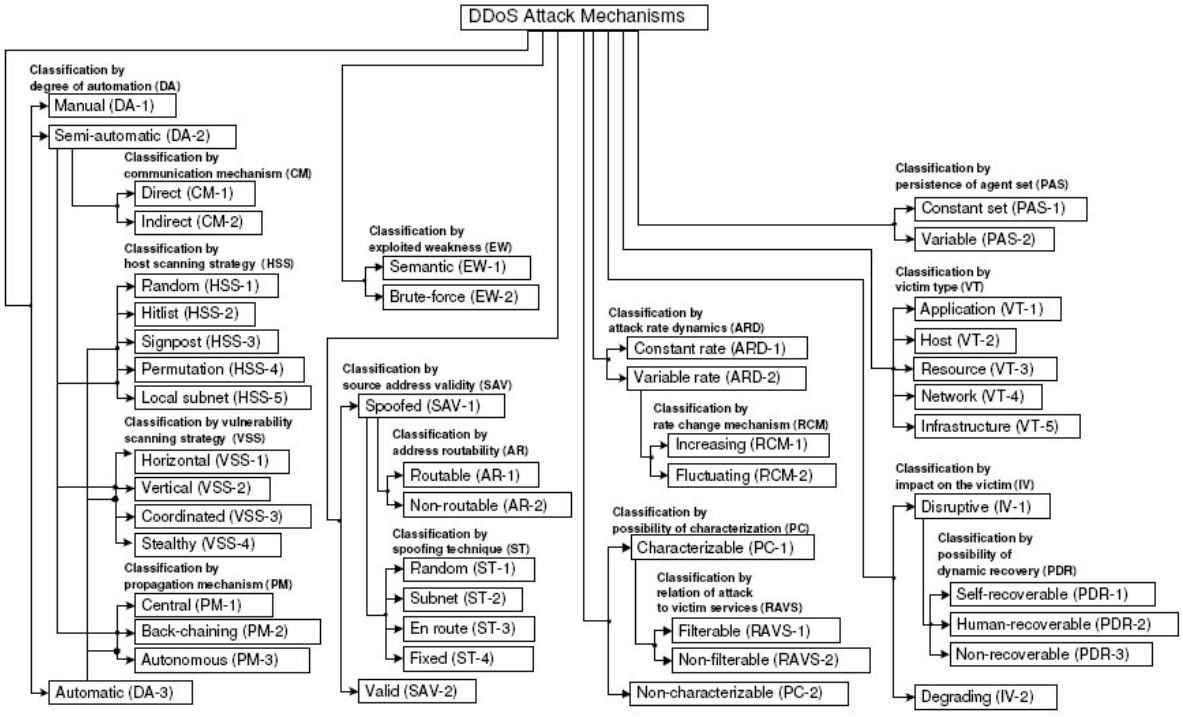

Figure 1.3: Taxonomy of DDoS attack mechanisms [17]

can be executed, and the different results that it can have. It shows that a DDoS attack can be classified by the degree of automation, how it exploits the weakness, source address validity, attack rate, and persistence of agent set. Figure 1.3 shows that the results of a DDoS attack can be classified by victim type and impact on the victim. For instance, a DDoS attack may affect only a host, or possibly the entire network. The way it affects the host may be degrading, in which it uses up parts of its victim's resources, or flat out disruptive, in which recoverability levels differ. Recovery levels should be an important characteristic to look at when determining the impact. However, the taxonomy does not provide a link between attack name or execution type and the consequence.

Lindqvist and Jonsson [14], along with the Department of Computer Engineering at Chalmers University of Technology in Sweden, made a first attempt at correlating attack type with consequence. In their paper, they use attack technique classifications originally created from Neuman and Parker, refer to [14] for more information on this. In order to

correlate attacks to consequences, they utilized 25 undergraduate students who were taking a course in applied computer security. The test facility consisted of 24 SUN ELC diskless workstations, all connected to one file-server. Each student accurately recorded their attacks and the administrators set up proper monitoring software to capture all activities and breaches. Figure 1.4 shows the attack results and is a sub-set of the entire attack technique taxonomy presented in [14].

Category			Number of intrusions
NP5 Bypassing intended controls	Password Attacks	Capture	6
		Guessing	12
	Spoofing privileged programs		6
	Utilizing weak authenticaion		13
NP6 Active misuse of resources	Exploiting inadvertent write permission		12
	Resource exhaustion		0
NP7 Passive misuse of resources	Manual browsing		1
	Automated searching	Using a personal tool	0
		Using a publicly available tool	8

Figure 1.4: Sub-set of the taxonomy of intrusion techniques [14]

The resulting taxonomy, shown in the left hand pane of Figure 1.4, is classified by the three traditional results of computer security: confidentiality, availability, and integrity. Breeches in these three classifications in turn result in exposure, denial of service, and erroneous output, respectively. Each of these have sub-categories that Lindqvist and Jonsson believe the first two levels are appropriate for all networks. The third level may be more specific to network system like their own test bed.

Figure 1.5 shows the correlation of attacks to the resulting consequences, and thus is the basis of a framework for mapping attack techniques to consequences. Notice that ”Bypassing intended controls, password attack by guessing,” (NP5-pg), was executed 12 times during the intrusion experiment, and always resulted in increased access to a user’s account. However, some attacks did not result in a one-to-one mapping. The extreme case shows that a ”Bypassing intended controls, by utilizing weak authentication,” (NP5-uwa) resulted in a mapping to five different sub-classifications in two different taxonomy classes. It appears that the Chalmers University of Technology has been working on this problem

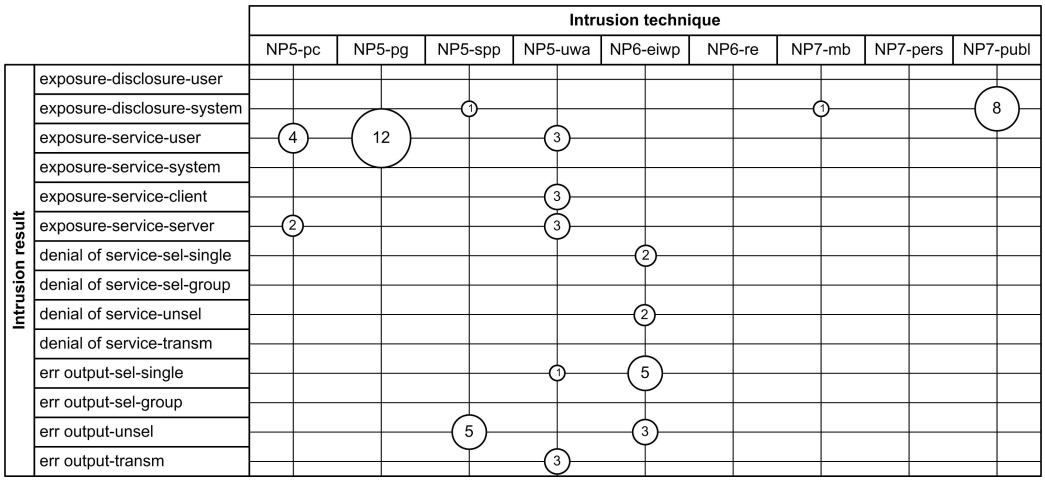

Figure 1.5: Distribution of intrusions according to technique and result [14]

since 1997, however it was difficult to find more current results.

Although feasible, the task at hand is extremely large and would take time and resources only available to a large corporation. This is simply due to the large range of attacks and vulnerabilities that exist. A variety of organizations have taken on the role of providing up to date information about attacks and vulnerabilities. A few of these include CVE [18], Bugtraq [5], US-CERT [28], and Snort [25].

1.4 VTAC Approach

This thesis develops an impact assessment framework, VTAC: Virtual Terrain assisted impact Assessment for Cyber attacks. The design of VTAC is separated into two main components: a virtual terrain and impact assessment algorithms. The virtual terrain models the network, its characteristics, and relationships necessary to perform an impact analysis. Once created, impact assessment can be performed by overlaying attack data onto the virtual terrain and processing it.

Chapter 2 defines the virtual terrain and discusses how it could potentially be automatically populated. The developed impact algorithms are presented in Chapter 3, followed by the test network and simulation environment used to test VTAC in Chapter 4. Presentation

and insight into the results are made in Chapter 5. Finally, Chapter 6 summarizes what was accomplished and offers possible extensions to this work.

Chapter 2

Virtual Terrain Definition and Generation

This chapter will focus on the virtual terrain that was developed to model networks and their configuration. The automatic generation of the virtual terrain model and the schema that can be used to define the terrain are also discussed.

2.1 Preliminary Design Thoughts

The goal in creating a virtual terrain model is to be able to store, gather, and update the information and characteristics of a computer network necessary to perform impact assessment. These characteristics are discussed in the following sections.

A reasonably secure network typically has multiple access domains. Direct access to these internal and often critical domains or subnets is limited or prohibited. Serious cyber attacks, thus, need to exploit different system vulnerabilities and progress through multiple domains. Reasoning on the progress made by a cyber attack shall benefit from a contextual model - a virtual cyber terrain that models the logical accessibility from one access domain to another. Most importantly, the cyber terrain should model the system and network configurations, including their vulnerabilities that may be exposed as the attacker compromises one or more systems in the network.

Once this model is created and populated, it will be not only be used for storing the

current state of the network, but also for analysis by impact assessment or other algorithms. When developing the virtual terrain model, the usage of the terrain should be kept in mind so that the algorithms may efficiently use it.

2.2 Virtual Terrain Definition

A cyber terrain is modeled as a directed graph consisting of host, router, or user nodes that are interconnected with directed arcs. Both the nodes and arcs have attributes defined for impact assessment. The host node represents a machine or cluster of machines, the user nodes represent users and their accounts, and the router nodes represent connecting devices such as routers or switches. The attributes for each node may be found in Table 2.1, and are discussed further in the following subsections.

Within some of the nodes, there exists criticality values. Each of these criticality metrics are numerical values between 0 and 1 inclusive, defining the importance of the item to its parent node. A value of 0 means that the item is irrelevant to the operation of the parent, whereas a value of 1 corresponds to the items being of most value to its parent. A thorough study of how criticality may be defined to better assess the consequences of cyber attacks is needed. This work focuses on impact assessment of cyber attack actions and assumes that the numerical values are given.

Host	**Router**	**User**
A node identifier	A node identifier	User ID
IP address(es)	Router name	Account(s) with privilege level
Host name(s)	Neighbor Permission List	Account criticalities
Machine criticality	Allowed (Boolean)	
Neighbor list	Traffic flow permissions list	
Allowed (Boolean)		
Permission List		
Service tree(s)		
List of users		

Table 2.1: Summary of virtual terrain node attributes

Service Tree
Service name
Service criticality
Privilege level
Version(s) with privilege level
Exposures(s) with privilege level
Exposure damage score
Exposure CVE IDs

Table 2.2: Service tree attributes

2.2.1 Hosts

Notice that a single Host node may have multiple IP addresses to define a cluster of identically configured hosts or servers. This simplifies the terrain model as well as the impact assessment process. Machine criticality represents how important the operation of a host is to the overall network or a subnet of computers. Items that may affect this score may include, but are not limited to: what type of machine it is (workstation, File Server, Web Server, *etc.*), and how much volume of traffic it receives. The neighbor list on a host defines what other hosts and routers are directly connected it. Two attributes are defined with every arc: Allowed and Permission List. The Permission List contains a lists of protocols and IP addresses. The Allowed attribute is a Boolean value. When the attribute is 'true' the protocols and IP addresses on the Permission List are the only protocols and addresses allowed, and everything else is blocked. A value of 'false' defines the opposite.

Service Tree

Each Host node may have one or more services running; here we define service in a general term that includes both remote services and local user applications. A service tree is used to represent each service available in the node. Each service has a criticality value that corresponds to the importance of the service to its parent host. At the terrain model development phase, all running services will be scanned by tools to determine the open ports and vulnerabilities of each machine. This information then can be used to build the

service trees in each machine. To build these trees, one or more databases must be cross-referenced to determine which exposures and exploits should be mapped to each version of the running services. As also suggested by Massicotte *et al.*[16], we adopt Snort [25], Bugtraq [5], Nessus [10], and NMAP [11] to build a service database that will aid the creation of a service tree. The service tree will capture the name of the service, versions of that service that are running, and the IDS alerts that could be reported if the corresponding vulnerabilities were exploited.

A key feature of the service tree model presented here is that it captures privilege differences between services and between versions of the same service. Like regular users, every service runs at a given privilege level. While many services do run at the system level, other services can run only at the user level. If a service is exploited, the attacker usually gains access to the computer at the level of the service. In addition, services may be local or remote. A remote service can be compromised without obtaining access privilege to the computer hosting the service. Such services typically listen on a specified TCP/UDP port. A local service is a service that can only be exploited after the attacker gains access to the computer hosting the service.

The version(s) of each service are found on the second level of the tree. It is possible that multiple versions of the same service could be running on the same host. For example, a web programmer may wish to run multiple versions of Mozilla Firefox concurrently to test for the compatibility of a web site between different versions. Some vulnerabilities could be fixed or introduced from version to versions. Also, a specific version may also have a privilege different from that of its parent service. Therefore, our model allows the privilege to be defined explicitly for each version of each service. By default, if no privilege is defined, a version will inherit the privilege of its parent service.

The actual IDS alerts are the children of the versions. The IDS alerts classified under a service and version imply that if an IDS alert was reported on that host, the parent version of the service is affected by that alert. Like versions, these IDS alerts can also inherit

the privilege level of its parent. Some IDS alerts may not correspond to actual vulnerability exploits. For example, knowledge discovery attacks such as a TCP Syn attack may indicate that the target host is alive, but will not compromise any privileges to the hosts. Such alerts are defined with a 'None' privilege. The exposure damage score is used to determine the severity that an exposure may cause to a service given an alert. This damage score can be manually assigned by the security analysts or derived from different vulnerability scoring systems, such as Microsoft's proprietary scoring system [2], US-CERT [28], SANS [12], and the Common Vulnerability Scoring System (CVSS) [21]. CVSS is an open framework that provides equations for people (software vendors, security analysts, *etc.*) to prioritize risks associated with vulnerabilities across a common scoring scale. Besides the base vulnerability score, CVSS allows adjustments for temporal and environmental conditions. The temporal metric allows for adjusting vulnerability characteristics that change over time. The environmental metric represents changes to the vulnerability characteristics with respect to the network environment. Final CVSS scores range between 0 and 10 inclusive, with 0 meaning the vulnerability is harmless and 10 corresponding to a more lethal type.

The service trees provide a structural model to determine the extent to which services are compromised on each host. More specifically, it helps to determine the privilege(s) obtained by the attacker during the process of an attack. It also filters out false positive, *i.e.*, alerts that do not correspond to a service running on the target host or subnet. Furthermore, and perhaps more importantly, by correlating the services and privileges in different machines, the cyber terrain may be used to deduce potentially threatened targets with similar or the same running services. Inference using the services, however, depends on the connectivity between hosts and subnets.

2.2.2 Routers

In analyzing the progression of cyber attacks, the physical topology of the network is not entirely relevant. Routers and switches allow communications between hosts despite them

not being physically connected. The communications that take place between hosts, however, are subject to the configuration of the hosts themselves as well as the configuration of the routers and switches between the hosts. The attributes of the directed arcs in the cyber terrain are defined to capture these restrictions. Similar to the Host node, two attributes are defined with every arc: Allowed and Neighbor Permission List. Rather than containing access information to a single IP, the routers Neighbor Permission List defines traffic flow between a router's neighbors.

2.2.3 Users

Users typically have different types of accounts on different domains of machines. Accounts are used to identify the privilege level and purpose of hosts that the user can access. Each account has a criticality that defines how important the account is to its parent user. For example, workstations and servers may be put into separate accounts because they perform different tasks for a user.

2.2.4 VT Graphical Representation

Figure 2.1 shows a graphical representation of the virtual terrain described. The Host/Host Clusters are represented in the middle column as nodes with service instantiations as their children. The square children nodes of the services are the specific IDS alert exposures related to each service. The right side of the figure is each user and its associated accounts. The rectangular figures on the left side represent the routers. The hosts and routers form a bipartite graph in the rights side of the figure, with each user account mapping mapping to its respective hosts, and vice versa. The routers and hosts form a tree structure to show the path that traffic will physically flow between nodes. The solid lines represent the traffic flow between to nodes, while the dotted lines represent a physical connection, however it is not the path for traffic. Each of the firewall rule entries show the restrictions for traffic flow between each node.

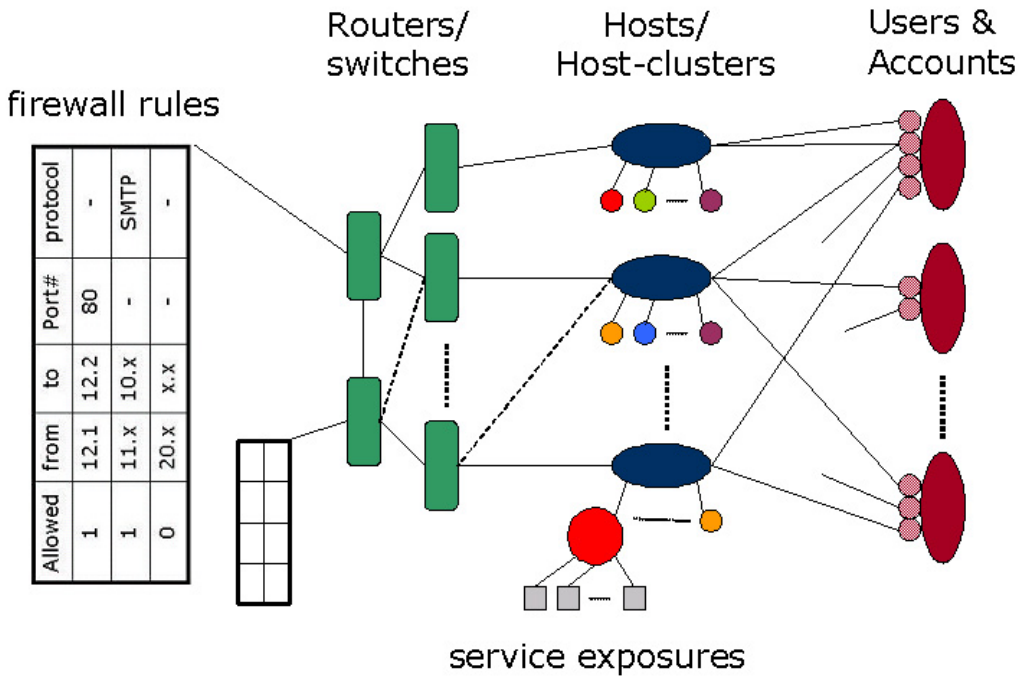

Figure 2.1: A graphical representation of the virtual terrain

2.2.5 Basic Virtual Terrain Implementation Details

The implementation of the virtual cyber terrain was done using Java 1.6.0 v1.0. An object-oriented approach was taken, keeping the structure similar to a network and its components. To help implementation go smoothly, a few assumptions were made without compromising the integrity of the model:

- All components are physically connected via a LAN. No wireless connections are used.
- A host only has one LAN connection, which must be connected to a router.
- A router may have multiple hosts and/or routers as neighbors.
- Only node assertions can be guaranteed to update correctly in real-time. To update other parts of the network, the terrain should be re-initialized.

The top level host, router, and user nodes are included in the .virtualterrain package, whereas the service tree levels were implemented in the .exposures package. The actual VirtualTerrain object was created simply to be a data structure with standard methods for getting and setting attribute values. An additional VTProcessor class, discussed in Section 3.3 can be used for gathering specific types of data or performing more in-depth actions on the VirtualTerrain object. Figures 2.2-2.4 can be referenced for the created objects and their methods used to implement the VirtualTerrain.

For more specific details about the virtual terrain implementation, please refer to the source code found on the thesis CD.

2.3 Generating the Virtual Terrain

The creation of a cyber terrain involves determining services running on each machine and cross-referencing those services with relevant vulnerabilities. Given the large variation in services that could run on a host and the large number of IDS alerts corresponding to vulnerabilities of the services, it is not realistic to manually create and update an accurate, complete cyber terrain for even a small network.

To automate the creation of a cyber terrain, a database mapping IDS alerts to susceptible services and versions is necessary. From the IDS alert alone, it is impossible to accurately determine what service was compromised by that attack. However, databases, e.g., Nessus and Bugtraq, provide information on which services (and versions) are susceptible to which vulnerabilities. In 2005, Massicotte, *et al.*[16], noted that 47% of Snort alerts did not provide Nessus or Bugtraq references, so those alerts need to be manually classified. Besides updating references, this would only be a one-time inconvenience.

Scanners such as Nessus and NMAP can be used to scan the network for the remote services running on each machine. Once the services are identified, the database discussed above can be queried for relevant IDS alerts and the service trees then can be created for each host. Nessus does provide CVSS scores, CVE IDs, and Bugtraq IDs for a select set

Figure 2.2: .virtualterrain package UML class diagram

Figure 2.3: .exposures package UML class diagram

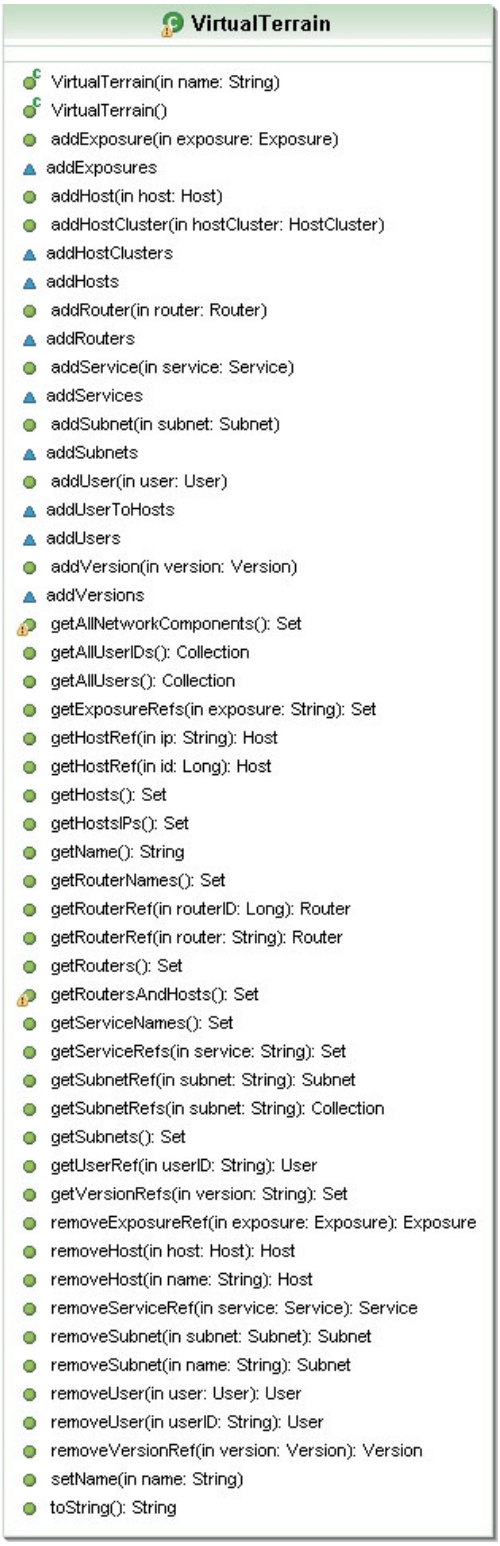

Figure 2.4: VirtualTerrain object UML class diagram

of vulnerabilities. The network scanning provides only remotely exploitable services. For some of these services, Nessus can provide user information along with Administrator, Guest, or User privilege level. Local services would need to be identified by the administrator or a local scan of each hosts.

The directed arcs representing allowed and banned protocol communication between hosts are critical to the terrain model. One possible way to define the arcs is by having each host scan all other hosts to determine remote access protocols or ports that are allowed or banned. NMap does have limited capability to penetrate poorly configured network obstructions such as firewalls. It could potentially be used to scan other hosts. This could, however, generate unwanted traffic. Alternatively, we could analyze or scan router and firewall configurations to determine the set of protocols allowed or banned between access domains or subnets. If the network contained dynamic firewall rules, update scans would have to be often run, or a system that monitors rule changes could be used.

From our research, it is unknown if criticality data can be gathered automatically. Defining the machine, service, and account criticalities is something that can be done initially by a network administrator.

Ideally, multiple scanners could be used to gather the necessary information and would all have the same output reports. A particularly interesting software package is NetMap, developed by the Reliable Software Group from the Department of Computer Science at the University of California Santa Barbara [31]. NetMap's purpose is to combine such common network scanning tools, making use of each tool's specialty, and use them to gain as much network information as possible. The results are then presented in a common format. If all of the necessary scanners did exist, a tool like NetMap could be used to automatically create the virtual terrain.

2.4 Virtual Terrain Schema

The focus of this work is to create a virtual terrain that can be used for impact analysis. Since a complete set of tools to gather all of the network information are not readily available, we create a virtual terrain schema that is used to define the virtual terrain. The schema is structured similarly to that of the virtual terrain, but has been slightly modified to ease the process of defining the virtual terrain model.

The schema is defined using the Extensible Markup Language (XML). XML provides an easy way to define nodes and their underlying attributes. An XML document is primarily made up of Elements and Attributes. Elements can be thought of as main objects and Attributes typically represent information about those objects. Figure 2.5 depicts a high level view of the virtual terrain schema, showing all of the Elements used to define the terrain.

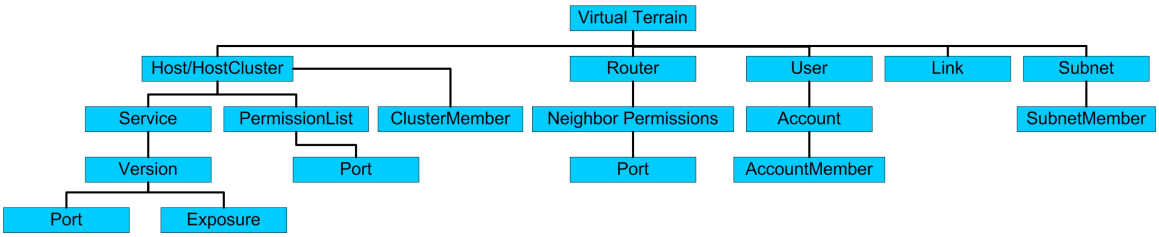

Figure 2.5: High level view of the virtual terrain schema

Notice that the Figure 2.5 shows Host/HostCluster, Router, and User elements that match to the nodes discussed in Section 2.2. Link and Subnet elements are additional to aid in defining physical network connections and the Router Neighbor Permission List. These two XML elements do not translate to specific objects in the virtual terrain, rather they add attributes to the Host and Router objects. The next subsection provides details on how the schema elements are translated to the virtual cyber terrain. The full schema is defined as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<VirtualTerrain>
  <User userID="">
    <Account name="" privilege="" account_criticality="">
      <AccountMember id="" />
    </Account>
  </User>
  <Host id="" ip="" name="" operation_criticality="" allowed="">
    <PermissionList ip="" >
      <Port port="" protocol="" />
    </PermissionList>
    <Service name="" privilege="" operation_criticality="">
      <Version name="" privilege="">
        <Port port="" protocol="" />
        <Exposure alert_signature="" privilege="" cvss_base_score="">
          <CVE prefix="" year="" id="" />
        </Exposure>
      </Version>
    </Service>
  </Host>
  <Router id="" ip="" name="" allowed="" >
    <NeighborPermissionList sourceID="" destID="" >
      <Port port="" protocol="" />
    </NeighborPermissionList>
  </Router>
  <HostCluster id="" name="" operation_criticality="" allowed="">
    <ClusterMember ip="" />
    <PermissionList ip="" >
      <Port port="" protocol="" />
    </PermissionList>
    <Service name="" privilege="" operation_criticality="">
      <Version name="" privilege="">
        <Port port="" protocol="" />
        <Exposure alert_signature="" privilege="" cvss_base_score="">
          <CVE prefix="" year="" id="" />
        </Exposure>
      </Version>
    </Service>
  </HostCluster>
  <Link sourceID="" destID="" />
  <Subnet id="" ip="" name="" >
    <SubnetMember id="" />
  </Subnet>
</VirtualTerrain>
```

2.4.1 XML Virtual Terrain Schema Reader

Java and JDOM (a Java API to manipulate XML) [13] were used to implement an object that can read in the schema and translate it to a VirtualTerrain object. The developed

functions parse and analyze the XML elements and attributes and use that information to create the virtual terrain. Figure 2.6 shows the UML diagram of the .io package that implements the XMLVirtualTerrainReader Object. The VTXMLElements object contains all of the constants used to define the schemas elements and attributes.

Most of the translation from the schema to the virtual terrain is self explanatory. Each Host/HostCluster and Router object have numerical unique identifiers that make populating the schema easier than providing specific names or IP addresses. Tables 2.3-2.6 provide details on the elements that may be unclear on how they are used.

One major item that has still not been covered is how the Internet will be defined. This is essential because the majority of attacks are outsider attacks, not insider. To model the Internet, a Host element can be used with the following configuration:

```
<Host id="0" ip="XXX.XXX.XXX.XXX" name="INTERNET"
operation_criticality="0.0" allowed="0" />
```

Once created, the components unique identifier can be used, via Link and Router Permission List elements to connect gateway routers directly to the Internet.

For more details on how this reader works, please see the source code located on the thesis CD.

Link
Purpose: To define a physical, two-way connection between network components.
1. The srcID may only be a Host/HostCluster or Router ID. 2. The dstID may be a Host/HostCluster, Router, or Subnet ID.

Table 2.3: Link element rules

Subnet
Purpose: To group a set of Host(s)/HostCluster(s) together that have the same neighbors and same permissions from their neighbors.
1. A subnet cannot contain another subnet. 2. A subnet cannot contain routers.

Table 2.4: Subnet element rules

Router Neighbor Permission List
Purpose: To define the traffic allowed or banned between a router's neighbors.
1. When allowed==1, if a component is not in the Permission List, the connection is considered closed. If a component is in the list, but empty, the connection is completely open. Else, the connection is limited to the ports and protocols in the list. 2. srcID and dstID can be a Host/HostCluster, Router, or Subnet ID.

Table 2.5: Router Neighbor Permission List element rules

Host Permission List
Purpose: To define the traffic allowed or banned to a host from a specific IP address.
1. When allowed==0, if a component is not in the Permission List, the connection is considered completely open. If a component is in the list, but empty, the connection is closed. Else, the connection is limited to the ports and protocols not on the list. 2. Component must be defined by IP address.

Table 2.6: Host Permission List element rules

Figure 2.6: .io package UML class diagram

Chapter 3

VTAC's Attack Processing and Impact Assessment Algorithms

This chapter will present both the attack processing and impact assessment algorithms used in VTAC. It will include a brief discussion about how the algorithms make use of the virtual terrain.

3.1 Attack Processing

The input to VTAC is IDS alerts. These alerts typically provide information such as source IP, destination IP, port, protocol, and an alert signature. Sometimes some of the information may be left out. For example, the port or protocol may be missing if the IDS cannot determine it. VTAC requires that the input be in the following format:

src<srcIP> **dst**<dstIP> **port**<port#> **protoc**<protocol> **alertSig**<alertSignature>

The alerts received by VTAC are assumed to have already gone through an alert correlator and false positive filter. As mentioned in Chapter 1, there are many existing systems and research projects whose primary focus is to correlate and filter alerts. Aside, it is well-known that false positives may still slip through these filters. Using the virtual terrain model and the attack data, VTAC can provide an additional layer of false positive filtering. Once processed, each attack will be labeled as either a logical or an illogical attack. To be

categorized as logical, all of the attack processing steps must be passed. Pseudo-code for the attack processing steps are displayed in Figure 3.1.

```
Boolean logicalAttack = false;
List validAttackPorts = {};
PortInfo givenPort = port/protocol from alert;
List<PortInfo> exposurePorts = {};

if( dst exists && dst.hasExposure(alertSig) ){
   //Get ports that attack could logically happen on.
   if( !givenPort.isEmpty() ){
      validAttackPorts = givenPort ∩ exposure.getPorts();
   else{
      validAttackPorts = exposure.getPorts();
   }

   //Ensure valid attack ports do not violate dst machine's personal banned list. Modify.
   validAttackPorts = validAttackPorts ∩ (ports ∉ dst.getBannedList(src));

   //Find the path for attack, and get allowed firewall rules along path.
   path = spanningTree(src, dst);
   pathRules = getFirewallRules( path );

   //Compare current set of ports against the paths firewall rules
   validAttackPorts = validAttackPorts ∩ pathRules;

   if( !validAttackPorts.isEmpty() ){
      logicalAttack = true;
   }
}
```

Figure 3.1: Pseudo-code for determining if an attack is logical

If the attack is logical, then it would seem as though the false-positive filters worked correctly. However, if the attack is illogical, there could be one of two problems: either the IDS alert is in fact a false positive, or there is an error in the way the virtual terrain is configured.

3.2 Impact Assessment Algorithms

To assess the impact of the attacks, a heuristic scoring scheme is developed to quantify the relative severity of damage on each network component. We define the impact scores for the hosts I_H, the services I_S, the users I_U, and the subnets or the entire network I_N as follows. Let H, S, U, and N be the set of hosts, services, users, and subnets (including the entire network), respectively. The lowercase letters shall represent the elements belonging to their corresponding sets. Let X act as a wild card for $H,\ S,\ U,$ or N. Note that multiple host nodes may run independent instances of same services. These instances are denoted as $e \in E$. In other words, if categorized in the type of service, E shall reduce to S. The combination rules that are used within the algorithms may not be the best, however they do allow us to demonstrate the concept of impact assessment.

3.2.1 Host Impact

Definition: The potential damage done to a host with respect to its services and their importance to the host.

To capture the current impact of a host, we analyze the services and the exposures that exist on that host. The impact score of a host (I_H), represented in Figure 3.2, is determined by looking at the importance of each service with respect to its parent host, and the damage score of the asserted exposures for each service. For each service on the host, the maximum asserted exposure damage score is retrieved and normalized to fit a 1.0 scale. The resulting score, α_e, represents the maximum damage that is imposed to the parent service:

$$\alpha_e \ = \ \max_{k \in K^*(e,t)} (\alpha_k); [0, 1] \tag{3.1}$$

Where α_k is the damage score assigned to the vulnerability exposure k and $K^*(e, t)$ is the set of asserted exposures associated with the service instance e at time t. The impact score for a host (Equation 3.2) combines each α_e score with its respective service criticality (c_e)

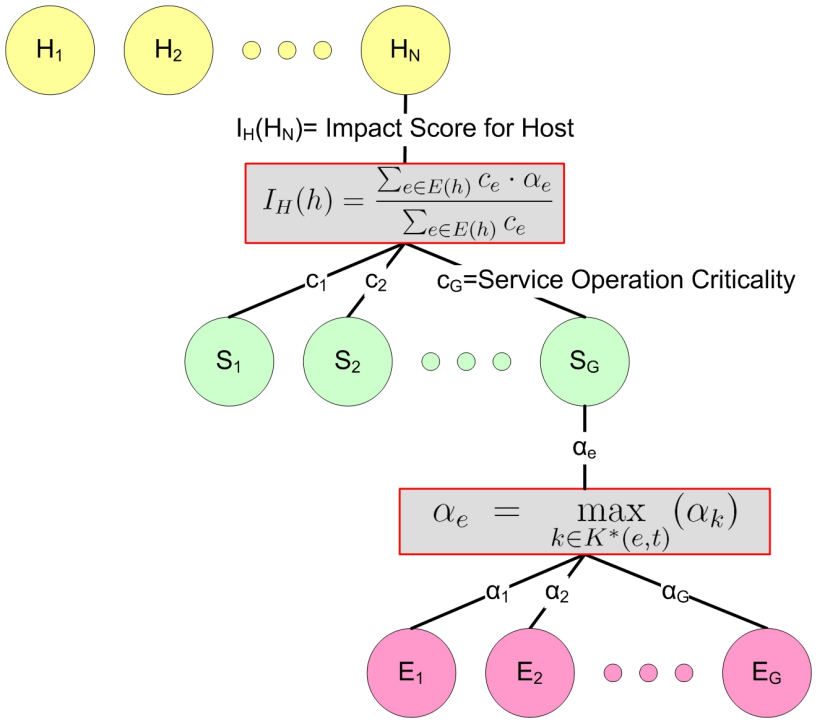

Figure 3.2: Impact score algorithm for a host

via weighted average:

$$I_H(h) = \frac{\sum_{e \in E(h)} c_e \cdot \alpha_e}{\sum_{e \in E(h)} c_e}; [0, 1] \tag{3.2}$$

This score will be important to analysts because it shows what hosts have been attacked, and the severity of the attack. The higher the I_H, the more damaged the host is considered.

3.2.2 Service Impact

Definition: How potentially compromised a particular service is over a given network.

To calculate the impact score for a service (I_S), every instance of a given service running in different hosts on the network is analyzed. This algorithm is represented in Figure 3.3. For each instance, the same combination rule is used as in the I_H calculation to get the maximum asserted exposure damage score, α_e (Equation 3.1). However, since in our terrain model we did not define an operation criticality score for services with respect to the overall network, we use a different combination rule to determine the I_S. Equation 3.3, a normal

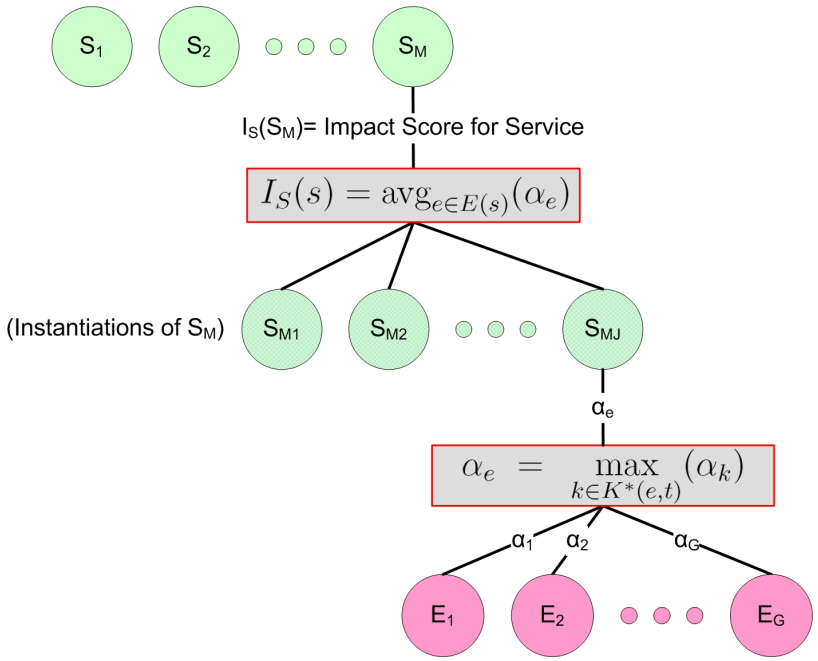

Figure 3.3: Impact score algorithm for a service

average of α_e scores, is used to define the I_S:

$$I_S(s) = \text{avg}_{e \in E(s)}(\alpha_e); [0, 1] \tag{3.3}$$

The impact score of a given service will give the network analyst an idea of how severely compromised a particular service is across the entire network. For example, assume there are two identical FTP servers in the network running a standard FTP service. If one of them is attacked, asserting an exposure with a damage score of 1.0, then the impact score of the FTP service would be considered 0.5.

3.2.3 User Impact

Definition: How potentially damaged the hosts are that a particular user has accounts on, with respect to the importance of those accounts.

User impact can be thought about as what affect attacks have on a user. The purpose of this score is to give the analyst a way to see what users have been affected by an attack.

It implies how badly compromised, or infected the machines that they use and access currently are. The impact score for a user (I_U), Figure 3.4, is determined by first looking at

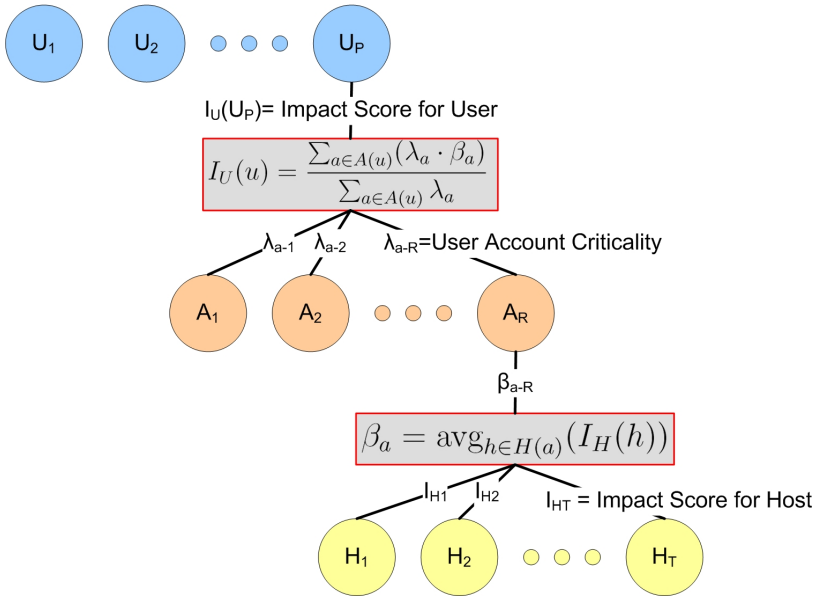

Figure 3.4: Impact score algorithm for a user

how severely compromised each host is within each account. To get a feel for this, we can use the normal average combination rule on the I_H for each account, Equation 3.4:

$$\beta_a = \text{avg}_{h \in H(a)}(I_H(h)); [0, 1] \tag{3.4}$$

Secondly, each of these accounts have different criticality scores with relation to its parent user. The weighted average of the β_a scores and the account criticality (λ_a) (Equation 3.5) is then used to determine the impact score for a user:

$$I_U(u) = \frac{\sum_{a \in A(u)} (\lambda_a \cdot \beta_a)}{\sum_{a \in A(u)} \lambda_a}; [0, 1] \tag{3.5}$$

3.2.4 Network Impact

Definition: How potentially damaged the entire network is with respect to the alerts received per its machines.

The network impact score will allow an analyst to monitor the healthiness of their network. The overall calculation of this score, shown in Figure 3.5, is primarily driven by each of the machines' operation criticality (η_h) with respect to the overall network, or for that matter, whatever group of machines that the analyst would like to monitor. Due to

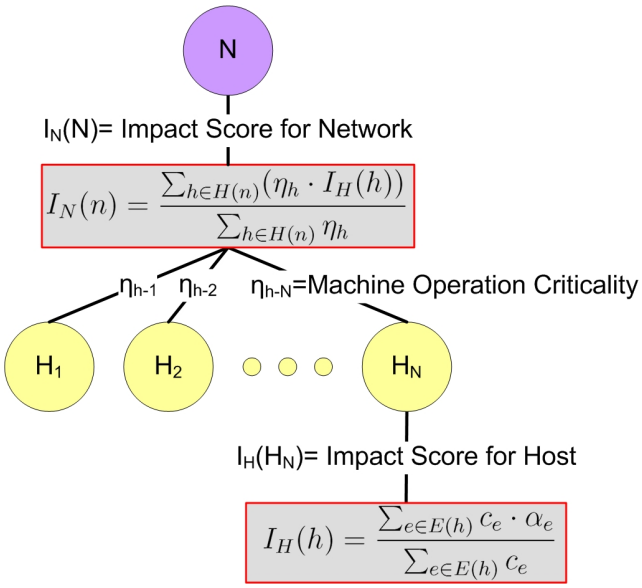

Figure 3.5: Impact score algorithm for the overall network

this, if the criticality values are properly assigned and machines are "zeroed out", this score could represent the impact on the overall network or a department within the network. The resulting impact score for the network (I_N), is calculated using the weighted average of host impacts in the virtual terrain (I_H) and machine operation critialities (η_h) (Equation 3.6):

$$I_N(n) = \frac{\sum_{h \in H(n)} (\eta_h \cdot I_H(h))}{\sum_{h \in H(n)} \eta_h}; [0, 1] \tag{3.6}$$

3.2.5 Reference Impact Scores

The four impact scores defined thus far, allow a network analyst to see the current damage done to the network. This may be sufficient to perform a rudimentary analysis, however it seems as though these scores are lacking a reference point to the potential damage that can actually be done. Since the current impact scores largely rely on exposure damage

scores and different criticalities, depending on the network configuration, impact scores may be incapable of reaching the maximum score of 1.0. To put the impact scores into reference with the current and potential situation, we propose two functions for each of the components being analyzed. These new reference scores will give an analyst a better overall view of the current situation. The two reference scores are defined as follows:

- **Impact score for X with Maximum Exposure for all Hosts (I_X-ME_H) -** Represents the highest possible impact score that a component could have, given that for all of the services related to it, regardless whether it is asserted or not, their exposure with the highest damage score is asserted.

- **Impact score for X with Maximum Exposure for asserted Services (I_X-ME_S) -** Represents the highest possible impact score that a component could have, if for all of the asserted services related to it, their respective exposure with the highest damage score is asserted.

The I_X-ME_H score can put the impact scores into perspective by showing the analyst the maximum damage that can be done to the X component. The calculation of I_X-ME_H for host, service, user, and network is similar as described in the previous sections, except for Equation 3.1. Instead, Equation 3.7 is used to calculate α, analyzing all existing exposures, not only the asserted ones.

$$\alpha_e = \max_{k \in K^*(e)} (\alpha_k); [0, 1] \tag{3.7}$$

Where this time, the iteration is done over $K^*(e)$, all exposures associated with the service instance e.

The score resulting from the second algorithm, I_X-ME_S, shows the analyst the maximum potential impact score given the currently asserted services. This results from an attacker fully exploiting the services of which they have already attacked, only on the hosts that they have already attacked. Figures 3.6 - 3.9 provide more details on how the I_X-ME_S scores are calculated for hosts, services, users, and networks. Notations $E(h)^*(t)$

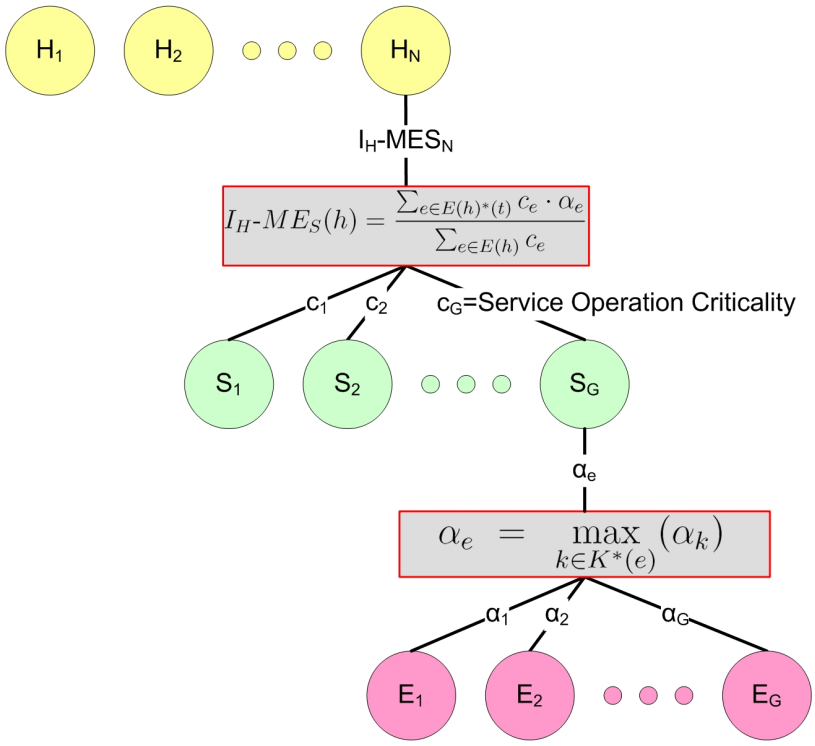

Figure 3.6: ME_S impact score algorithm for a host

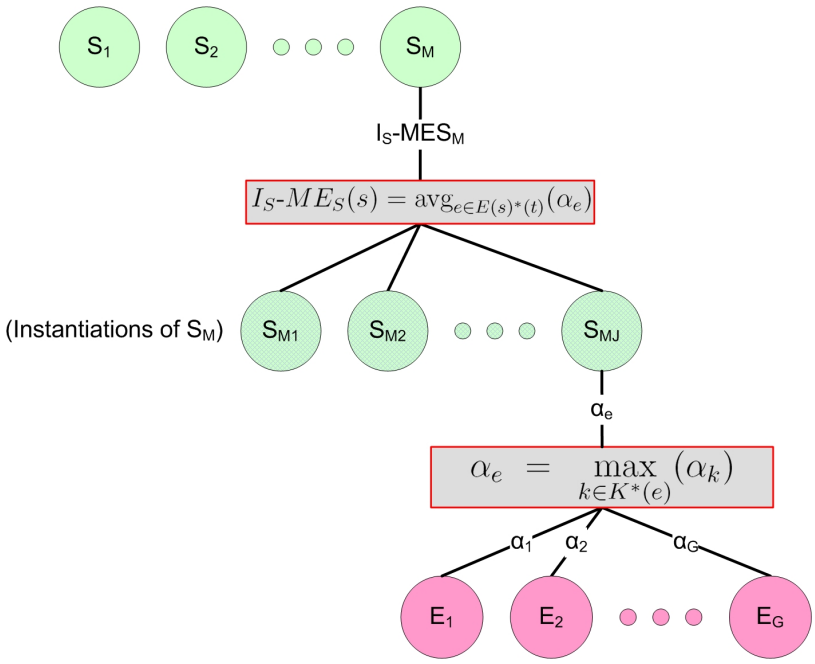

Figure 3.7: ME_S impact score algorithm for a service

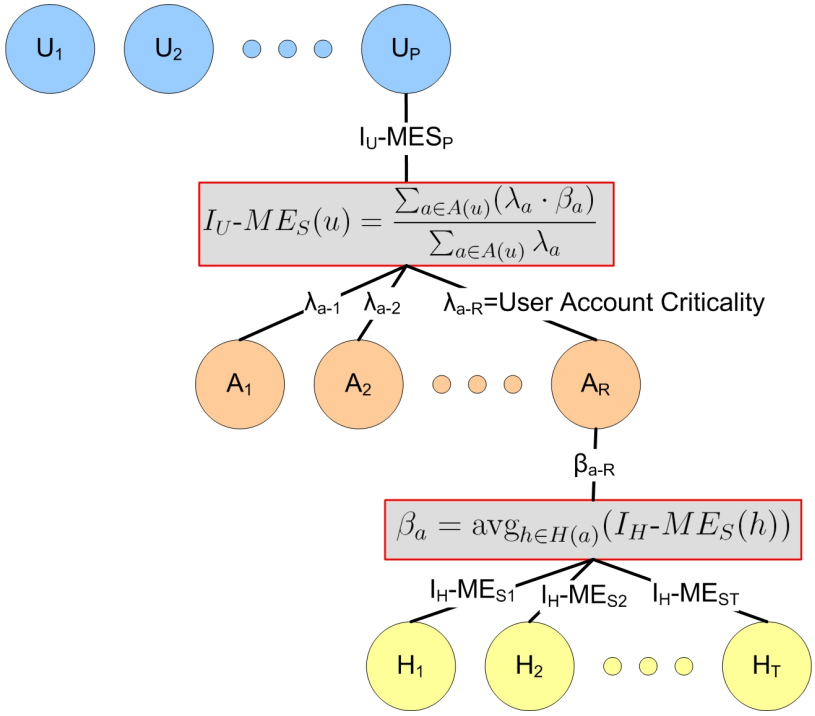

Figure 3.8: ME_S impact score algorithm for a user

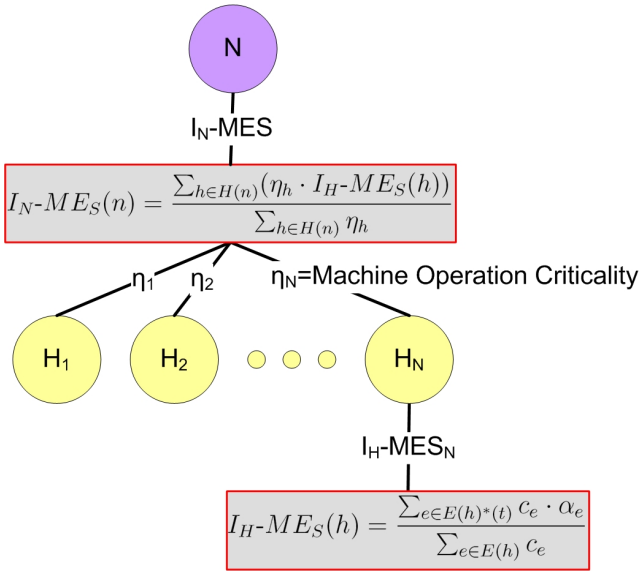

Figure 3.9: ME_S impact score algorithm for the network

and $E(s)^*(t)$ are used to represent the set of services on host h and the set of service instances of service s that are currently asserted at time t.

3.2.6 Projection

The algorithms presented in the previous sections assess the current impact that the attack may have. They do not provide projection capabilities of what the next attack actions could or will likely be. Knowing what an attacker is going to do next or what their real intent is, would be useful for an analyst. With this information they could take more appropriate actions to stop the attack, and limit both the harm to the network and the temporary inconvenience placed upon users for shutting down or closing connections on parts of the network. Another way to use this information could be to run simulated attacks to help identify the weaknesses of a network and common attack patterns.

The virtual terrain defined in Chapter 2 has the potential to be used for impact projection. A possible algorithm may include looking for machines that have the same exposures as those that have been already attacked. These machines can be checked against router configurations and the machines own firewall list to determine if an attack is even plausible given the current network setup. Another algorithm may find machines reachable from one hop, analyze their services, users, and overall importance to the network. These are just a few undeveloped ideas for projection algorithms using the virtual terrain model. They will not be discussed further in this thesis.

3.2.7 Computational Complexity with Virtual Terrain

The virtual terrain discussed in Chapter 2 was designed for algorithms to easily be able to use it. Ideally, once an attack is processed and the exposure is asserted, the affected components should be able to be retrieved quickly and only their impact scores should be updated. This is in fact how the algorithms operate, with the components being looked up with a key to a HashMap, which is performed in constant time, $O(c)$. Once the algorithm

has the desired object, it can now traverse up or down its parent/child tree (Section 2.2) to gather the necessary objects to perform the impact algorithms on. Given the destination IP and alert signature, Table 3.1 shows the pseudo-method-calls needed to get the components from the virtual terrain (vt) to be analyzed for impact assessment on a per attack basis. Once these objects have been retrieved, the algorithms discussed in sections 3.2.1 - 3.2.5 can be executed.

Impact Type	Pseudo-code
Host	vt.getHost(dstIP);
Service	vt.getHost(dstIP).getService(vt.getExposure(alertSig).getVersion().getService().getName());
User	vt.getHost(dstIP).getUsers();
Network	vt.getAllHosts();

Table 3.1: Pseudo-method-calls for retrieving objects to perform impact analysis on per attack

3.3 Basic Implementation Details

To keep a modular design, the implementation of both the attack processing and impact assessment algorithms is done separately from the virtual terrain. This allows future attack processing or assessment algorithms to be easily added, without disrupting existing functions.

The VirtualTerrain class from Section 2.2 was primarily designed as a data structure, not a data processor. Therefore, the VTProcessor class was created to hold the variety of functions used to do in depth processing of the virtual terrain. The idea is that any algorithm that would be using the virtual terrain could make use of this class. The implemented function signatures can be seen in Figure 3.10.

To process attacks and implement the specific impact assessment algorithms, the AttackProcessing class, Figure 3.11 was created. This class stores and controls all of the impact score data and controls attack processing as discussed in Section 3.1. To aid in executing and organizing these algorithms, the AttackProcessing class makes use of the VTProcessor class for detailed data retrieval or processing of the virtual terrain model.

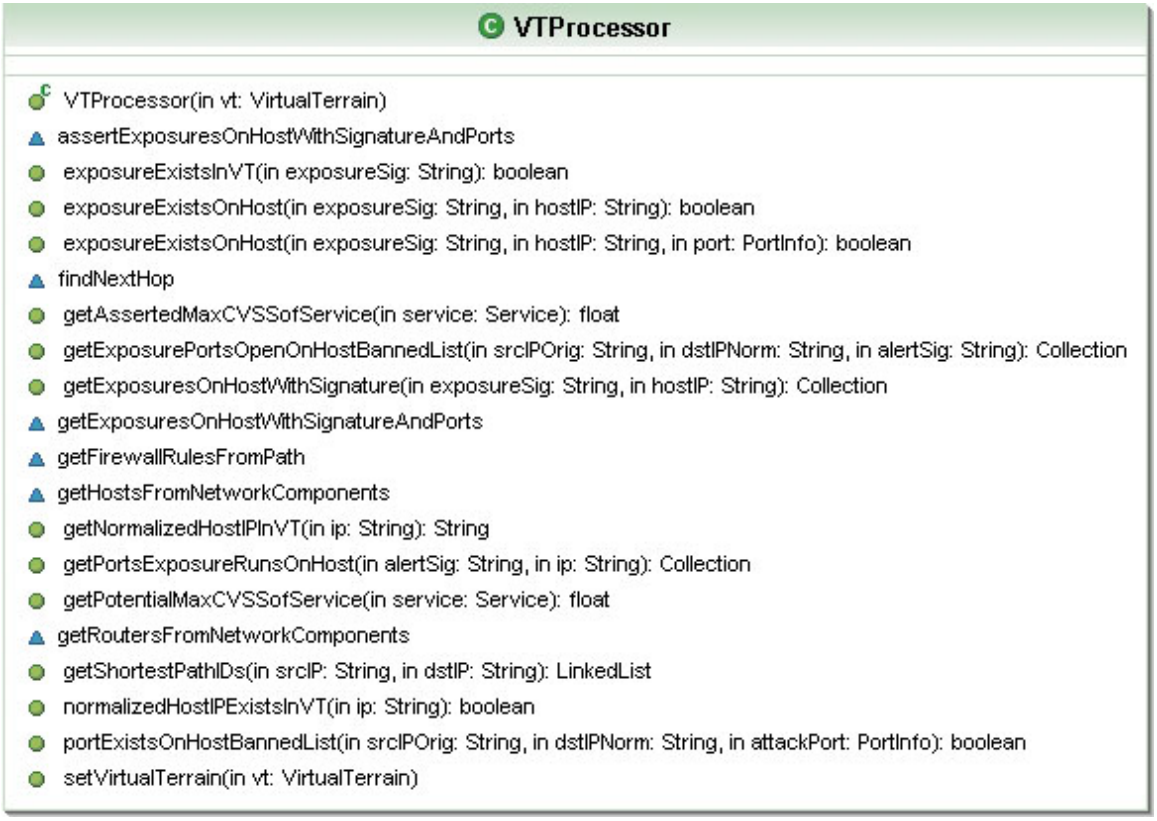

Figure 3.10: VTProcessor UML class diagram

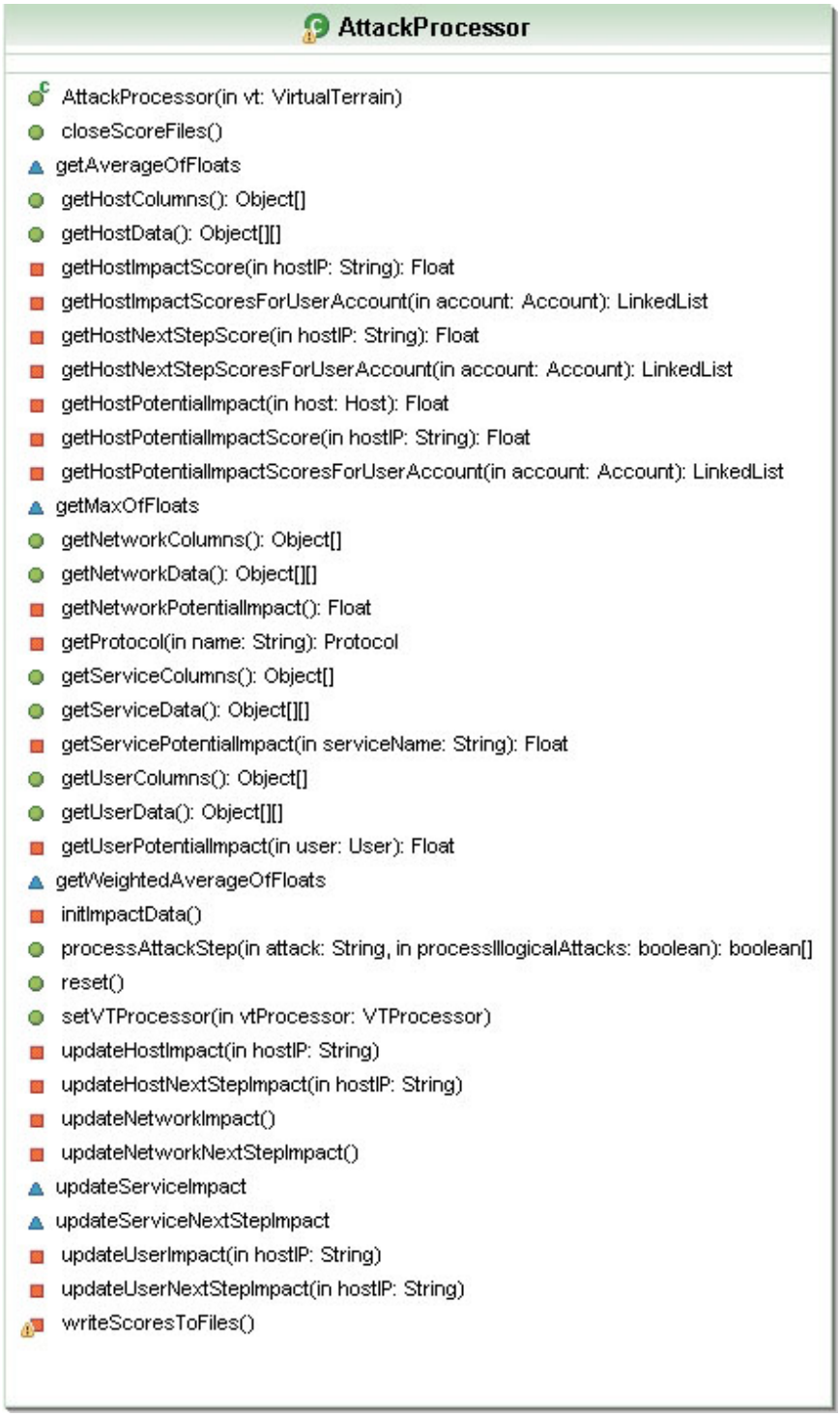

Figure 3.11: AttackProcessing UML class diagram

Chapter 4

Simulator and Test Network Configuration

This chapter will discuss how the attack experiments were conducted, how data was collected, and the test network that the experiments were run on. In Section 4.1, the software developed to simulate the attacks and impact assessment algorithms will be presented, followed by Section 4.2 which introduces the test network and its configuration, developed to run the attack data and impact algorithms.

4.1 Attack and Virtual Terrain Simulator

In Chapters 2 and 3, the framework and algorithms to perform impact assessment were introduced. A tool capable of running attacks is needed, showing the current status of the virtual terrain, and displaying or collecting the different impact scores for hosts, services, users, and the network.

Developed using Java 1.6.0 v1.0, the GUI shown in Figure 4.1 is used to drive the attack experiments. This GUI was developed strictly for testing and experimental purposes, so it may not have all of the features or display the data in the way a commercial product would. It has three windows, each with its own purpose:

- **Information Window -** Displays the *current* status of the virtual terrain. Individual

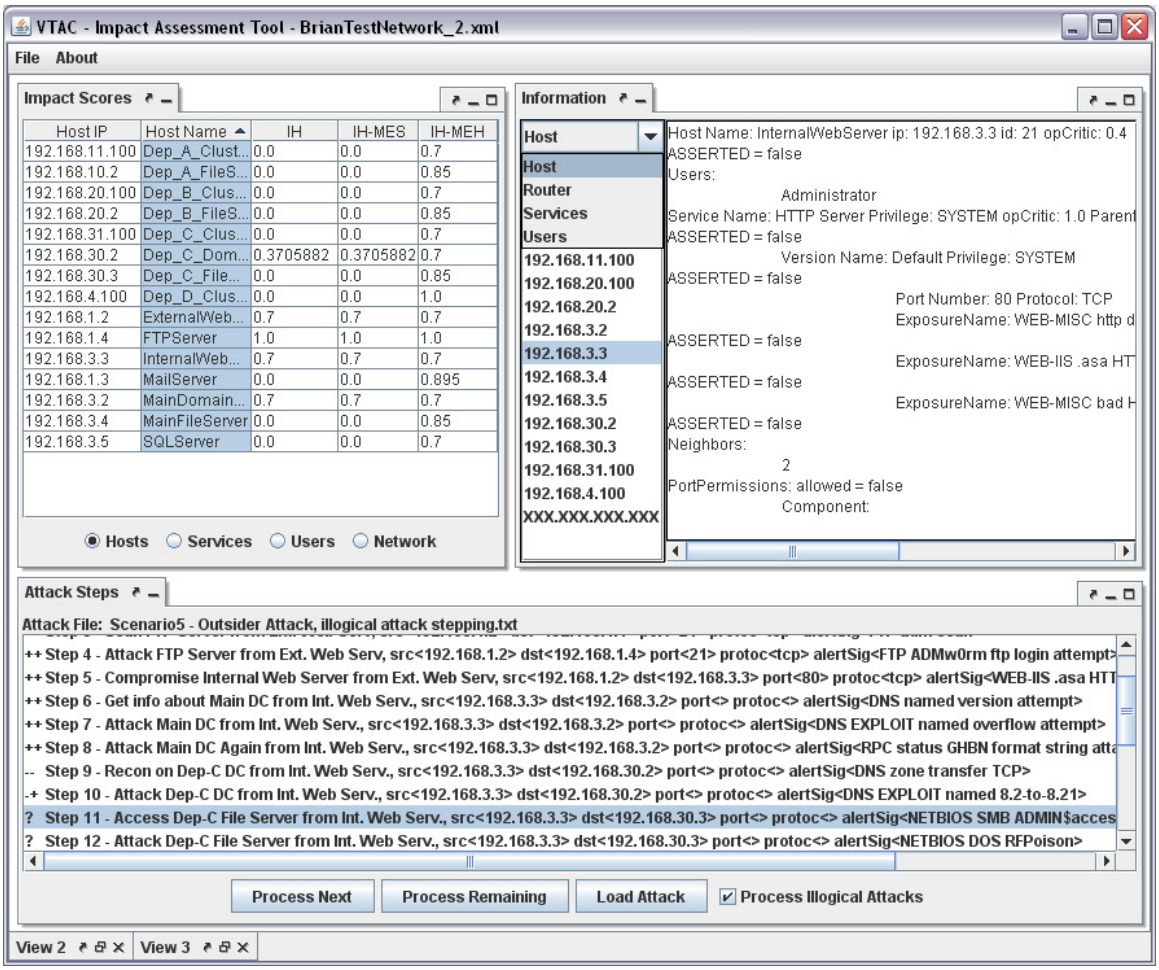

Figure 4.1: Screenshot of the impact assessment tool GUI in the middle of an attack.

hosts, routers, services, and users can be selected to display all of the relevant information for that object. This informs the user as to what exposures, services, and hosts have been asserted by an attack.

- **Impact Scores Window -** Displays the *current* impact scores for hosts, services, users, and the network in a table. The table can be sorted ascending or descending with respect to any column.

- **Attack Steps Window -** This window drives the simulator. It is primarily used to load and process attacks, but also controls whether or not to have VTAC fully process illogical attacks, as discussed in Section 3.1. There can be three different labels on an

attack once it enters VTAC: ++ (logical attack, attack processed), -+ (illogical attack, attack processed), or - - (illogical attack, attack not processed). Processing an attack means to assert the respective exposure.

Since the Impact Scores Window only displays the current impact score, and does not keep track of past scores, data collection can become difficult, especially for attacks with many steps. To help ease the data collection process, an impact score writer has also been implemented that records the impact scores for each component every time an attack step is executed. Each group of scores is output to a semi-colon delimited file that can easily be imported and processed in Microsoft Excel.

The source code for the attack simulator GUI can be found on the thesis CD.

4.2 Test Network

To examine and validate the performance of the virtual terrain and impact assessment algorithms, a test network needs to be created. Ideally, this network would be similar to an enterprise network, and cyber attack data would be publicly available for testing. From the research done on this project, we have not found that these items are readily available to the public. Valeur, Vigna, Kruegel, and Kemmerer [29] discuss that there is a lack of networks with full configurations needed to provide impact assessment. For VTAC, criticality metrics are a key item for the impact scores. These criticalities essentially map to the relationships between network assets such as machines, services, users, and vulnerabilities. By not having these relationship definitions, there is a large hole in the data needed to conduct impact assessment. Also, network infrastructures support different missions which are nearly impossible to interpret, especially without the asset relationships. Valeur, *et al.*, also mention the lack of real world attack data sets and what their purposes are. Initially the focus of these data sets was on testing low-level attack detection systems and now it has shifted to high-level attack analysis. Even so, these new data sets are more focused on testing alert correlation systems, not impact assessment systems. Regardless, without a

fully defined network (including mission statements) to go along with the correlated alerts these data sets would not provide us much value.

After thorough discussions with network security specialists and IT personnel, a mock network was created and configured to the best of our knowledge to help demonstrate impact assessment. Attack scenarios are created from IDS alerts to simulate different types of attack. These scenarios are assumed to be output from an alert correlator such as INFERD [26], [27]. They will be presented in Chapter 5 along with the results.

A few comments should be made about the design and configuration of this network and networks in general. First, even if a network is secured as tightly as possible and has all of the correct configurations, if the applications running on the network have vulnerabilities, the network can still be intruded. Secondly, it is not uncommon for networks to have security holes resulting from lackadaisical configurations. Lastly, and most importantly, the purpose of this network is to test the concept of impact assessment and to validate the algorithms. For this reason alone, we want a variety of network configurations to be able to test, most of which should have security holes that can be exploited.

4.2.1 Network Configuration

Figure 4.2 shows the terrain information associated with the test network. It was designed using a layered network approach, with each successive layer capable of having choke points to restrict traffic flow. Typically on a layered network, traffic is limited flowing down the network, but is free to travel up the network. By creating server or demilitarized zone (DMZ) subnets "out-of-band," this prevents traffic that is flowing up and out of the network from accessing the servers. Figure 4.3 summarizes the router configurations (traffic flow) on a neighbor-to-neighbor basis. The table gives the services that have access from the top row to the left column IP address.

For further details on the network configuration not found in the following subsections, please refer to the XML file on the thesis CD that defines the configuration of this network.

Servers

The 192.168.1.X subnet provides public services to external users, which include an external web server, a mail server, and an FTP server. Subnet 192.168.3.X includes all of the internal services, including an internal web server, an SQL server, a domain controller, and a file server. The internal web server is one that can be accessed network wide and functions as this network's intranet. The centralized domain controller is the point of authentication for most of the network, and the main file server serves as a data repository for some of the departments. Each server has at least some of the necessary remote services running on the machine required to provide that service.

Departments

The network was initially setup with four departments. To aid in examining impact data and creating interesting situations, each of them has a different configuration and plays a unique role. Each department, which all contain a cluster of workstations, may or may not have access to the centralized domain controller and file server. Depending on their configuration, they may have their own group of servers on a separate subnet, or even on the same subnet. Table 4.1 summarizes the key aspects of each department. Department

	Authentication		**File System**		**Other Access**			
Department	Main	Internal	Main	Internal	Mail	FTP	Int. Web	Internet
A	✓	-	-	✓	✓	✓	✓	✓
B	✓	-	✓	✓	✓	✓	✓	✓
C	-	✓	-	✓	✓	-	✓	✓
D	✓	-	✓	-	✓	✓	✓	✓

Table 4.1: Summary of department configurations

C is locked down relatively tightly, because it has its own internal file server and domain controller, so there is no need for file or authentication traffic to be allowed through to its department. Comparatively, Department B is relatively exposed. Although it has its own file server, it also has access to the main domain controller and file server, thus permitting

file and authentication traffic. For more detailed information about what type of traffic is permitted, please see Figure 4.3.

Users

Figure 4.2 indicates the users and the machines they have accounts on. The services running on each machine are also displayed. The users are assigned accounts to specific department workstations and also have accounts on machines that they subscribe to, *e.g.*, a file or FTP server. A global Administrator exists that has accounts on every machine in the network. Departmental administrators also exist. Most regular users only have access to one department. However both Jon and Kate have access to two departments. Jon has access to Departments A and B and Kate has access to Departments B and C.

4.2.2 Populating the Network

The test network created does not physically exist, therefore Nessus, NMap, or other scanners could not be used. The network characteristics were manually entered into the virtual terrain schema.

The machine, service, and account criticality values along with the exposure damage values were arbitrarily assigned. Values were entered that seemed reasonable for the mission of the network infrastructure. For example, for the machine criticalities, servers are generally given higher values because they play a more important role in the overall network. Depending on what type of server, and the location (main or department specific server), the criticality could also be affected. Service criticality values are determined by how important that service is to that respective machine. The FTP service that an FTP server provides is obviously more important to its functional purpose than a telnet or remote media service. Exposure damage scores were populated with respect to the type of attack they are associated with. Table 4.2 shows an outline of the damage scores assigned to each exposure. Notice that the impact algorithms from Chapter 3 do not take into account

Type of Attack	Damage Score
Reconnaissance	0
Intrusion (User)	4
Intrusion (System)	7
Escalation	7
Goal	10
Miscellaneous	1

Table 4.2: Summary of populating exposure damage scores

privilege level. The separation of categories compensates for this. Account criticality values were based on how important the machines on that account are for the user. For normal users, Table 4.3 is used to outline how important machines typically are. Administrator's

Machine Type	Account Criticality
Domain Controller	0.8
File Server	0.5
FTP Server	0.1
Mail Server	0.3
Workstation	1.0

Table 4.3: Summary of populating normal user account criticalities

account scores are similar to that of a normal user, except that they are given a high criticality for the machines that they are directly responsible for.

Both Tables 4.2 and 4.3 act as guidelines to assign the respective scores. All scores may not follow the tables exactly. For more details on what criticality scores were assigned, refer to the XML schema definition on the thesis CD.

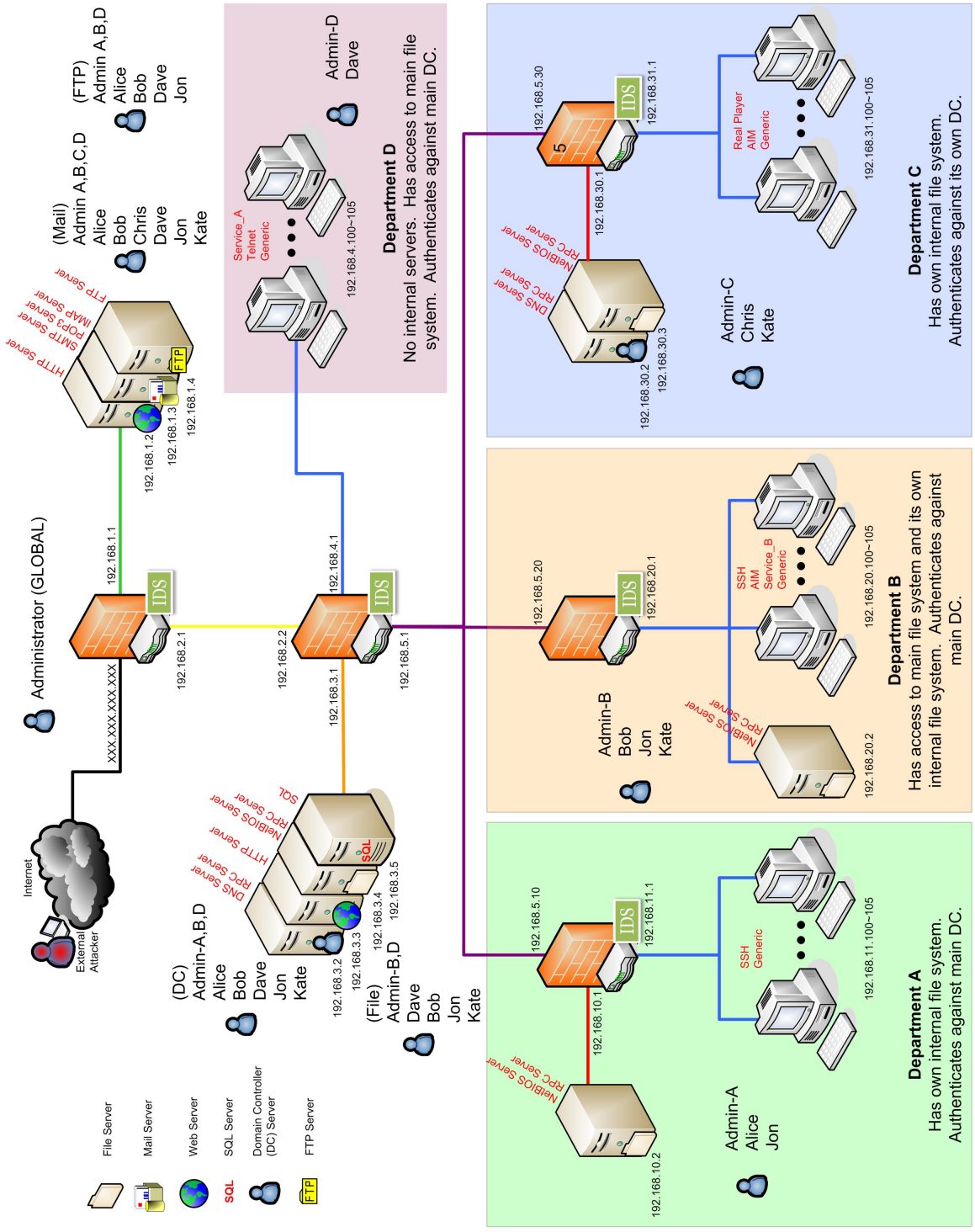

Figure 4.2: Test network showing machine services and user accounts

	FIREWALL TRAFFIC			Internet	192.168.1.1	192.168.2.1	192.168.2.2	192.168.3.1	192.168.4.1	192.168.5.1	192.168.5.10	192.168.5.20	192.168.5.30	192.168.10.1	192.168.11.1	192.168.20.1	192.168.30.1	192.168.31.1
			RID	0	1	1	2	2	2	2	3	4	5	3	3	4	5	5
		RID	NID	0	10 11 12	2	1	20 21 22 23	200	3 4 5	2 4 5	2 3 5	2 3 4	30	300	400	50 51	500
Internet	**Internet**	**0**	**0**	X	OPEN	OPEN	X	X	X	X	X	X	X	X	X	X	X	X
R1 Green	**192.168.1.1**	**1**	**10 11 12**	SMTP Webmail WEB FTP	X	POP3 IMAP WEB FTP	X	X	X	X	X	X	X	X	X	X	X	X
R1 Yellow	**192.168.2.1**	**1**	**2**	WEB SSH REALPLAYER AIM	POP3 IMAP WEB FTP	X	*OPEN*	X	X	X	X	X	X	X	X	X	X	X
R2 Yellow	**192.168.2.2**	**2**	**1**	X	X	*POP3 IMAP WEB*	X	OPEN	OPEN	OPEN	X	X	X	X	X	X	X	X
R2 Orange	**192.168.3.1**	**2**	**20 21 22 23**	X	X	X	POP3 IMAP WEB FTP	X	FILE DC Web SQL	FILE DC Web	X	X	X	X	X	X	X	X
R2 Blue - Dep D	**192.168.4.1**	**2**	**200**	X	X	X	POP3 IMAP WEB TELNET SERV-A GENERIC	OPEN	X	Closed	X	X	X	X	X	X	X	X
R2 Purple - Dep A-C	**192.168.5.1**	**2**	**3 4 5**	X	X	X	POP3 IMAP WEB SSH REALPLAYER AIM	FILE DC WEB SSH GENERIC	Closed	*X*	*POP3 IMAP DC WEB*	*POP3 IMAP File DC WEB*	*POP3 IMAP WEB*	X	X	X	X	X
Dep A Purple	**192.168.5.10**	**3**	**2 4 5**	X	X	X	X	X	X	*POP3 IMAP DC WEB*	*X*	OPEN	OPEN	OPEN	OPEN	X	X	X
Dep B Purple	**192.168.5.20**	**4**	**2 3 5**	X	X	X	X	X	X	*POP3 IMAP FILE DC WEB*	OPEN	*X*	OPEN	X	X	OPEN	X	X
Dep C Purple	**192.168.5.30**	**5**	**2 3 4**	X	X	X	X	X	X	*POP3 IMAP WEB*	OPEN	OPEN	*X*	X	X	X	OPEN	OPEN
Dep A Red	**192.168.10.1**	**3**	**30**	X	X	X	X	X	X	X	(2) POP3 IMAP WEB	X	X	X	FILE	X	X	X
Dep A Blue	**192.168.11.1**	**3**	**300**	X	X	X	X	X	X	X	(2) POP3 IMAP DC WEB (2,4,5) SSH GENERIC	X	X	OPEN	X	X	X	X
Dep B Blue	**192.168.20.1**	**4**	**400**	X	X	X	X	X	X	X	X	(2) POP3 IMAP FILE DC WEB (2,3,5) SSH GENERIC AIM	X	X	X	X	X	X	X
Dep C Red	**192.168.30.1**	**5**	**50 51**	X	X	X	X	X	X	X	X	X	POP3 IMAP WEB	X	X	X	X	FILE DC
Dep C Blue	**192.168.31.1**	**5**	**500**	X	X	X	X	X	X	X	X	X	(2) POP3 IMAP WEB REALPLAYER GENERIC	X	X	X	OPEN	X

NO CONNECTION
INTO FIREWALL
OUT OF FIREWALL

Figure 4.3: Table showing the configuration of the network access lists on a neighbor basis

Chapter 5

Results and Discussion

The main goal of this research is to introduce the capabilities of the virtual terrain and to present results from the developed impact assessment algorithms. The emphasis of this chapter is not on the actual performance of the algorithms, rather to show that impact score trends correctly align with the attacks presented and how network analysts may interpret or use the results. Eight different attack scenarios will be reviewed, discussing key aspects of each one and potentially how a network analyst could use this information to advance their knowledge of the attacks.

5.1 Impact Scores Illustrated via Scenario 1

Figure 5.1 illustrates a fairly simple cyber attack and can be used to demonstrate the basic capabilities of the impact assessment algorithms. Each of the red arrows and attack bubbles identifies the steps in an attack, showing where the attack is coming from and the victim machine. Table 5.1 gives a summary of the steps and the corresponding flagged IDS alert. The next few sub-sections show impact score results at every attack step, and briefly discuss how they follow the intended patterns.

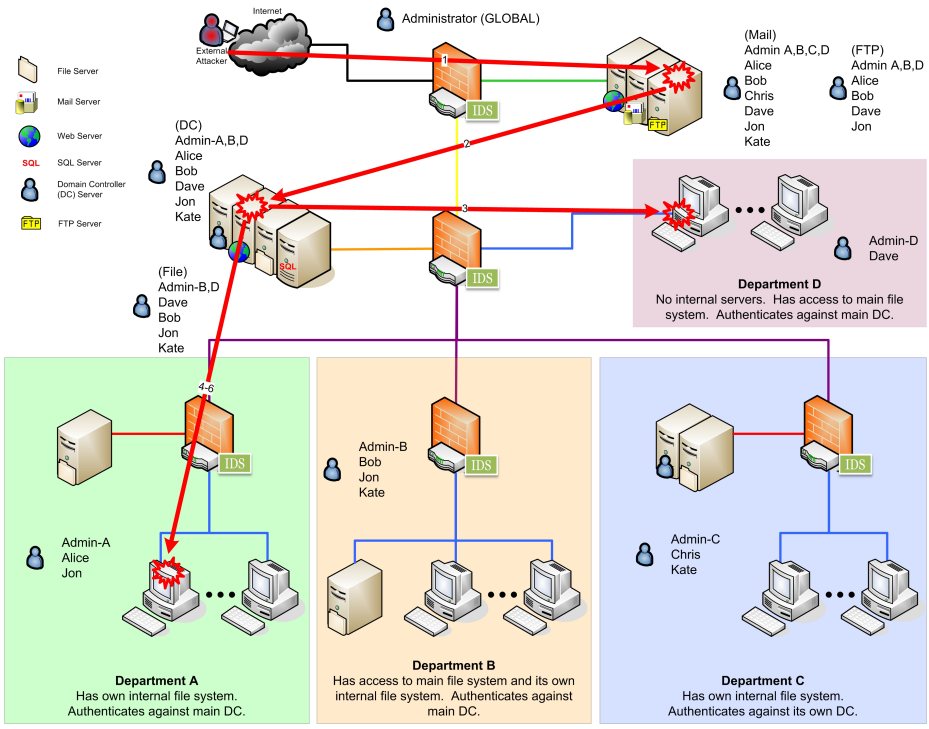

Figure 5.1: Topological view of scenario 1's attack steps

Step	Description	Source IP	Dest IP	Port/Prot	Alert Signature
1	Compromise FTP Server	129.21.168.101	192.168.1.4	21/tcp	WEB-MISC /home/ftp access
2	Step to Internal Web Server	192.168.1.4	192.168.3.3	80/tcp	WEB-IIS .asa HTTP header buffer overflow attempt
3	Attack Department D Cluster	192.168.3.3	192.168.4.100	23/tcp	TELNET bsd telnet exploit response
4	Ping Dep A Cluster	192.168.3.3	192.168.11.103	456/icmp	ICMP PING Microsoft Windows
5	Scan Dep A Cluster	192.168.3.3	192.168.11.103	22/tcp	SCAN SSH Version map attempt
6	Attack Dep A Cluster	192.168.3.3	192.168.11.103	22/tcp	EXPLOIT ssh CRC32 overflow

Table 5.1: Scenario 1's attack steps

5.1.1 Host Impact

Figure 5.2 displays the impact scores for each host per attack step. According to Chapter 3, a host's score should update when an exposure on one of its services is asserted. The graph in Figure 5.2 shows that the impact scores for attacked hosts changing on the correct step. Notice that on step 1, the FTP server's I_H changes, on step 2, the internal web server's score changes, on step 3, Department D's cluster I_H changes, and finally on step 6, Department A's cluster I_H changes. The alerts in steps 4 and 5 are considered reconnaissance, and have damage scores of 0, thus not affecting the host's I_H.

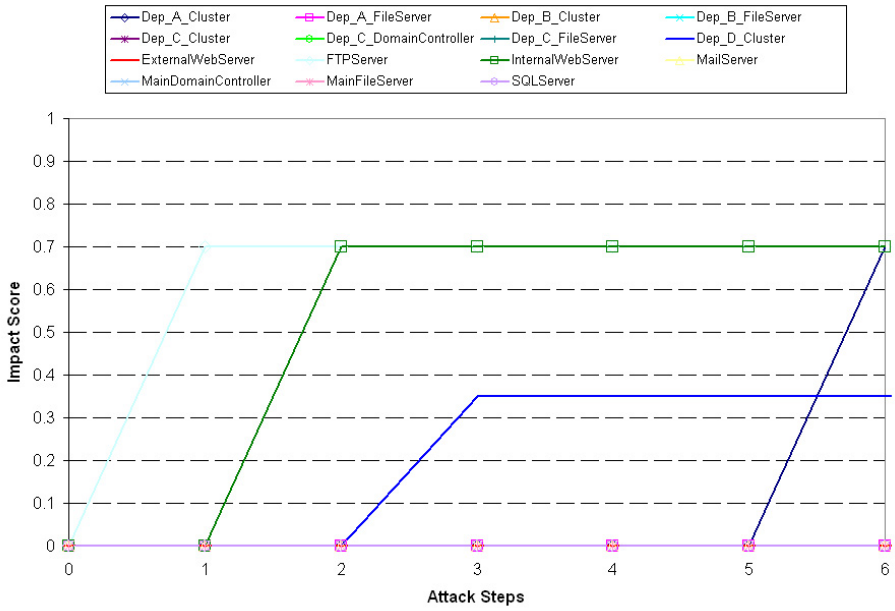

Figure 5.2: Host impact scores for scenario 1

5.1.2 Service Impact

In Figure 5.3, the impact scores for each service are shown changing for each step in the attack. A service's impact score is expected to change under the similar conditions as the host impact score. Again, referencing the algorithms from Chapter 3, the I_S should change when an exposure on one of the service's instantiations within the network has been asserted, and if it has a larger damage score than any other asserted node in that particular instantiation. The graph shows the correct services changing with respect to the attack. The FTP service is first asserted in step 1, followed by the HTTP service in step 2, the Telnet service in step 3, and the SSH service in step 6. Once again, the alerts in steps 4 and 5 do not affect the I_S results.

5.1.3 User Impact

The impact scores for each user per attack step are presented in Figure 5.4. Algorithms from Chapter 3 show that a user's impact score should change whenever an exposure has

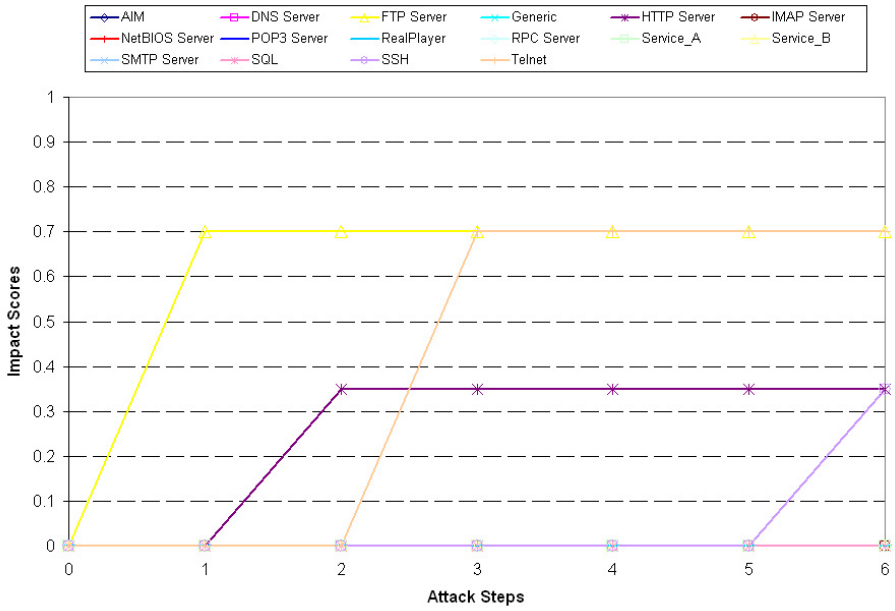

Figure 5.3: Service impact scores for scenario 1

been asserted causing a change in any of the host impact scores that the user has an account on. The first observation made is that there is a lot more movement in this graph than in Figures 5.2 and 5.3. This is primarily due to the fact that a single machine typically supports multiple users, therefore all of them will be affected when attacked. This can be easily pointed out in Figure 5.4 on step 6 when the Department A Cluster is attacked. All of the users who have accounts on that cluster, Administrator, Admin-A, Alice, and Jon all have their impact scores changed. Each of them change differently depending on how many accounts they have and how important the account with Department-A Cluster is to them.

5.1.4 Network Impact

Figure 5.5 displays the results for the impact score of the network (I_N). According to our definition, the I_N score is expected to change whenever a I_H score changes in the network. The graph follows the correct pattern, as it changes at steps 1,2,3 and 6, but not 4 or 5 as these are steps where no host impact scores changed (Figure 5.2).

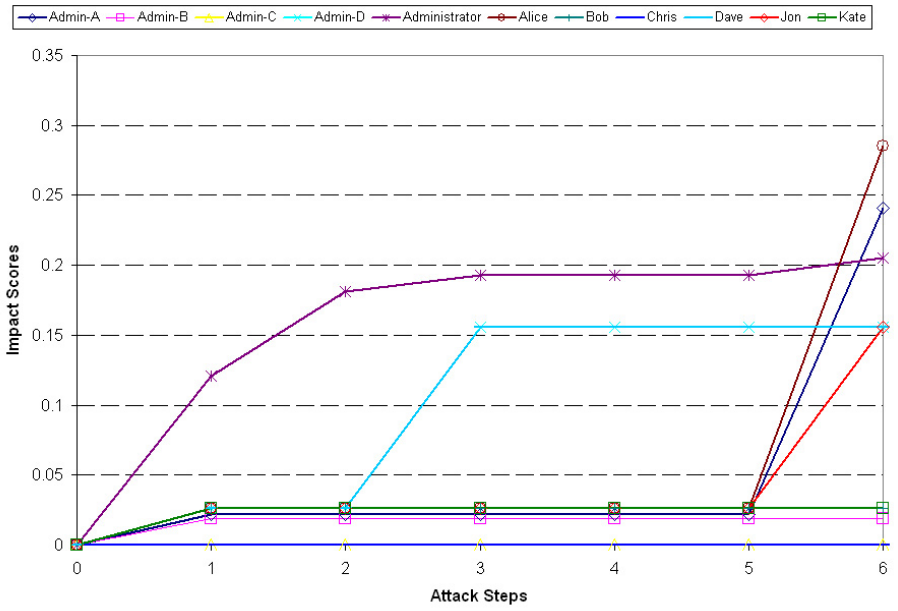

Figure 5.4: User impact scores for scenario 1

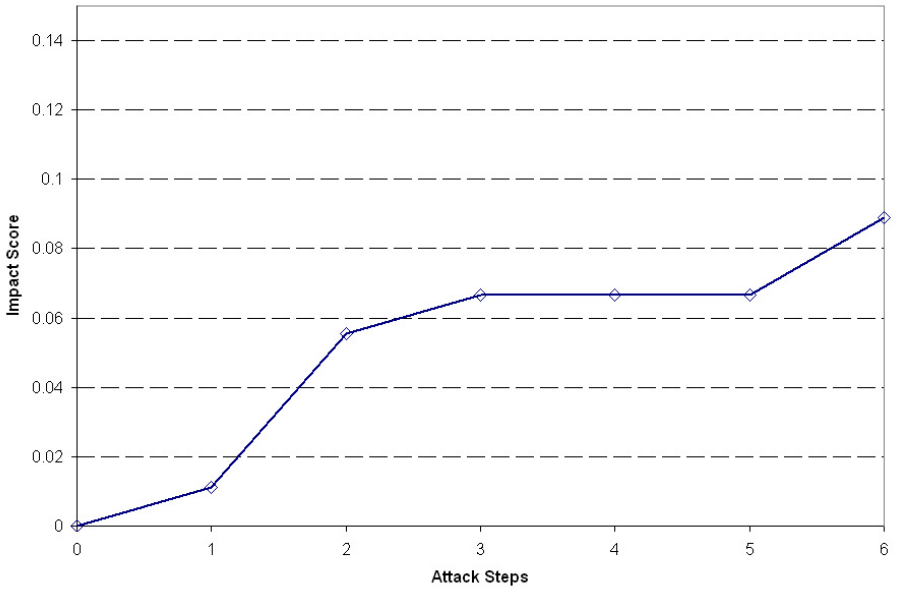

Figure 5.5: Network impact scores for scenario 1

5.1.5 Reference Scores

The two reference impact scores discussed in Chapter 3, I_X-ME_H and I_X-ME_S can also provide valuable insight for an analyst, especially when compared to the regular impact scores. However, displaying all three of the impact scores for each component on the same plot would make them very cluttered and difficult to read. Ideally, a single plot would be able to display all three impact scores simultaneously so that the analyst could easily study the situation.

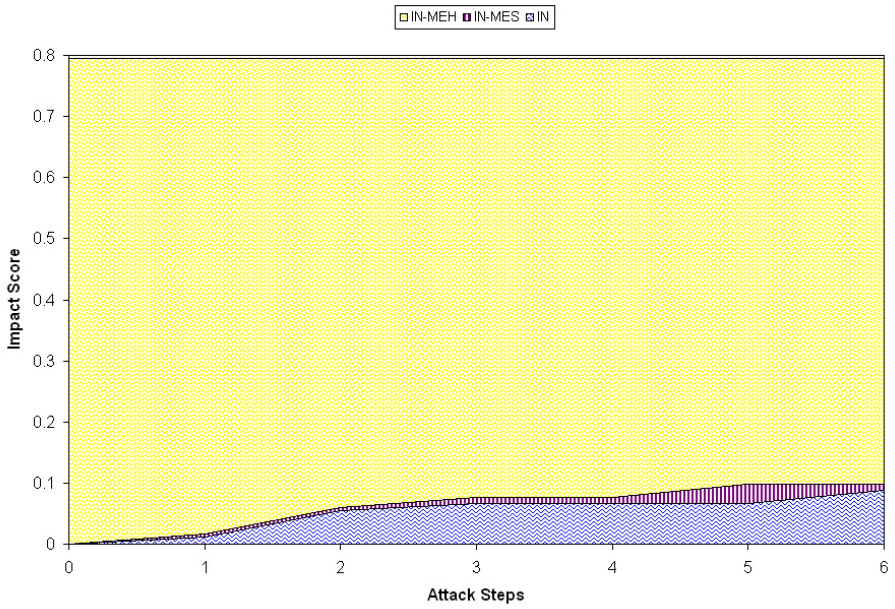

Figure 5.6: Impact scores for the network for scenario 1

For the network, these scores can easily be displayed as shown in Figure 5.6. Notice that there are three separate areas showing on the plot. The I_N area displays a score representing how much of the network has potentially been impacted by attack. The difference between the I_N and I_N-ME_S line is the I_N-ME_S envelope area. This area shows what the impact could be if the already asserted services were fully exploited, which could be very useful to indicate the impact of the security flaw of the corresponding services. The remaining I_N-ME_H area represents the maximum impact score that can be reached for the network. Using the graph to visually compare this versus the I_N and I_N-ME_S allows the analyst to

easily assess the current and potential impact on the network.

Unless a particularly interesting situation arises, the scenarios that will be presented in the following sections will only display the standard I_X impact scores. Note that the I_X-ME_H scores are constant for a given configuration. Tables 5.2-5.5 summarizes the I_X-ME_H scores for the entities in the test network.

Hostname	I_H-ME_H
Dep_A_Cluster	0.7
Dep_A_FileServer	0.85
Dep_B_Cluster	0.7
Dep_B_FileServer	0.85
Dep_C_Cluster	0.7
Dep_C_DomainController	0.7
Dep_C_FileServer	0.85
Dep_D_Cluster	1.0
ExternalWebServer	0.7
FTPServer	1.0
InternalWebServer	0.7
MailServer	0.895
MainDomainController	0.7
MainFileServer	0.85
SQLServer	0.7

Table 5.2: I_H-ME_H scores for the test network

Service Name	I_S-ME_H
AIM	0.4
DNS Server	0.7
FTP Server	1.0
Generic	0.0
HTTP Server	0.7
IMAP Server	0.7
NetBIOS Server	1.0
POP3 Server	1.0
RealPlayer	1.0
RPC Server	0.7
Service_A	1.0
Service_B	1.0
SMTP Server	1.0
SQL	0.7
SSH	0.7
Telnet	1.0

Table 5.3: I_S-ME_H scores for the test network

User ID	I_U-ME_H
Admin-A	0.7745313
Admin-B	0.7847298
Admin-C	0.7631818
Admin-D	0.87166667
Administrator	0.81551737
Alice	0.7605556
Bob	0.76055557
Chris	0.7513462
Dave	0.87166667
Jon	0.76055557
Kate	0.7605557

Table 5.4: I_U-ME_H scores for the test network

Network	I_N-ME_H
Test Network	0.7954762

Table 5.5: I_N-ME_H scores for the test network

5.2 Scenario Analysis

The following sections discuss a variety of scenarios, each of which may add to the overall impact assessment analysis.

5.2.1 Scenario 2

Scenario 2, displayed in Figure 5.7 is a relatively simple attack that starts by compromising the external web server. Next, it attacks the FTP server and then steps to the internal servers by attacking the internal web server. From there the attacker applies a back door to the Department D cluster and also compromises the main DC. Finally, the attack targets Department B's file server followed by Department A's cluster and file server. Detailed attack steps can be found in Table 5.6.

Looking at the I_H results in Figure 5.8, a network analyst can tell that through step 4, only machines in the external server domain may have been attacked. The I_N results in Figure 5.11 show that only attacking these machines has little affect on the overall network. Besides Administrator, no other users are affected until the FTP server is attacked. When the FTP server is attacked, users with accounts on occur an impact, however none of them

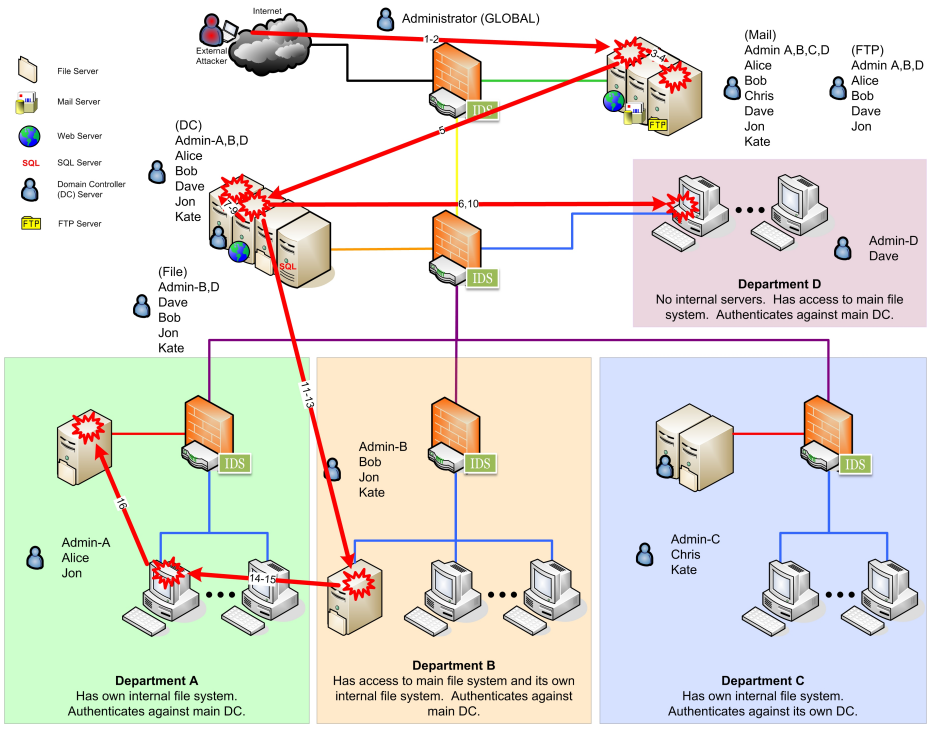

Figure 5.7: Topological view of scenario 2's attack steps

appear to be affected much (Figure 5.10). At this point, an analyst could use this information to help limit the potential damage. Since the FTP server really does not affect the overall network or users much, there should not be many problems that result from taking it off the network. The analyst could inform the users of the FTP server that any data on there may have been compromised and that it will be shutting for the time being. If necessary, they should use other means to transfer files.

As the rest of the attack progresses, impact scores increase relative to what has been attacked. The I_S shown in Figure 5.9 give the administrators a good idea of what services the attacker is compromising. This, combined with the I_H can be used to pinpoint the weaknesses in the network. Services whose score jumps multiple times on different machines are services that may be capable of bringing a network to its knees. Notice that the HTTP and NetBIOS service impact scores change a few times because two Web and File Servers are targeted on this specific attack. If this becomes a trend, it will become increasingly necessary to patch these services. A potential warning sign for future attacks on a particular

Step	Description	Source IP	Dest IP	Port/Prot	Alert Signature
1	Scan External Web Server	140.203.195.48	192.168.1.2	80/tcp	WEB-MISC http directory traversal
2	Attack Ext. Web Server	140.203.195.48	192.168.1.2	80/tcp	WEB-IIS .asa HTTP header buffer overflow attempt
3	Scan FTP Server	192.168.1.2	192.168.1.4	21/tcp	FTP adm scan
4	Attack FTP Server	192.168.1.2	192.168.1.4	21/tcp	FTP ADMw0rm ftp login attempt
5	Compromise Int. Web Server	192.168.1.2	192.168.3.3	80/tcp	WEB-IIS .asa HTTP header buffer overflow attempt
6	Ping Dep-D	192.168.3.3	192.168.4.100	456/icmp	ICMP PING Microsoft Windows
7	Get info about Main DC	192.168.3.3	192.168.3.2	/	DNS named version attempt
8	Attack Main DC	192.168.3.3	192.168.3.2	/	DNS EXPLOIT named overflow attempt
9	Attack Main DC Again	192.168.3.3	192.168.3.2	/	RPC status GHBN format string attack
10	Apply Backdoor to Dep-D	192.168.3.3	192.168.4.100	107/tcp	BACKDOOR subseven DEFCON8 2.1 access
11	Scan Dep-B File Server	192.168.3.3	192.168.20.2	/	NETBIOS SMB Startup Folder access attempt
12	Gain access to Dep-B File Server	192.168.3.3	192.168.20.2	/	NETBIOS SMB C$ access
13	Attack Dep-B File Server	192.168.3.3	192.168.20.2	/	NETBIOS DOS RFPoison
14	Scan for SSH on Dep-A Cluster	192.168.20.2	192.168.11.100	/	SCAN SSH Version map attempt
15	Attack Dep-A Cluster SSH	192.168.20.2	192.168.11.100	/	EXPLOIT ssh CRC32 overflow
16	Attack Dep-A File Server	192.168.11.100	192.168.10.2	/	NETBIOS DOS RFPoison

Table 5.6: Scenario 2's attack steps

service running on other machines may be if an attack on a service causes an I_H to rapidly increase, however the I_S is not particularly high (meaning that the service may be running on other machines as well).

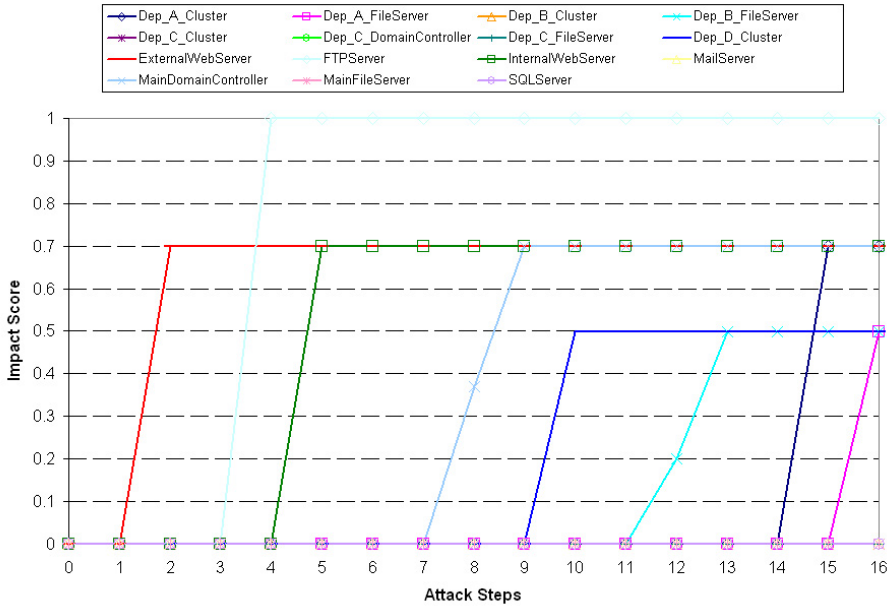

Figure 5.8: Host impact scores for scenario 2

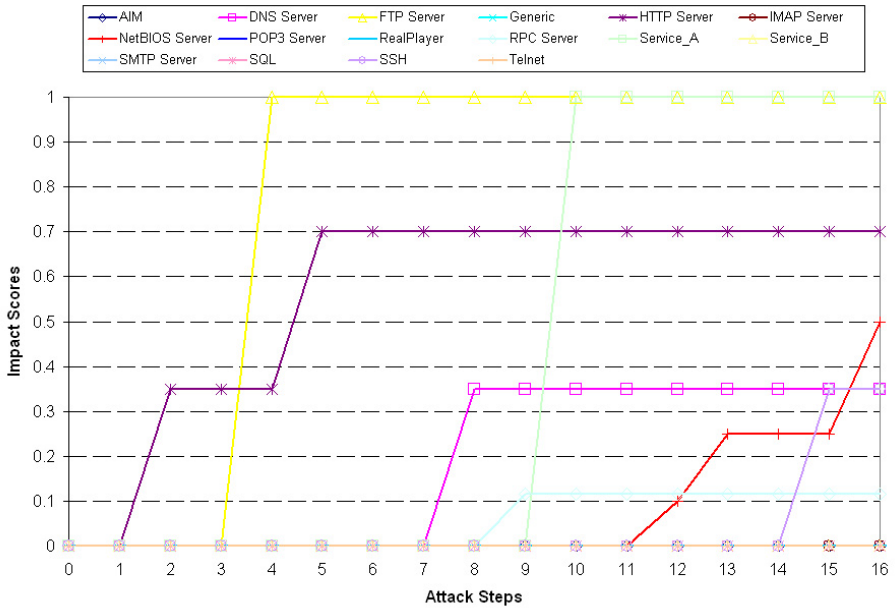

Figure 5.9: Service impact scores for scenario 2

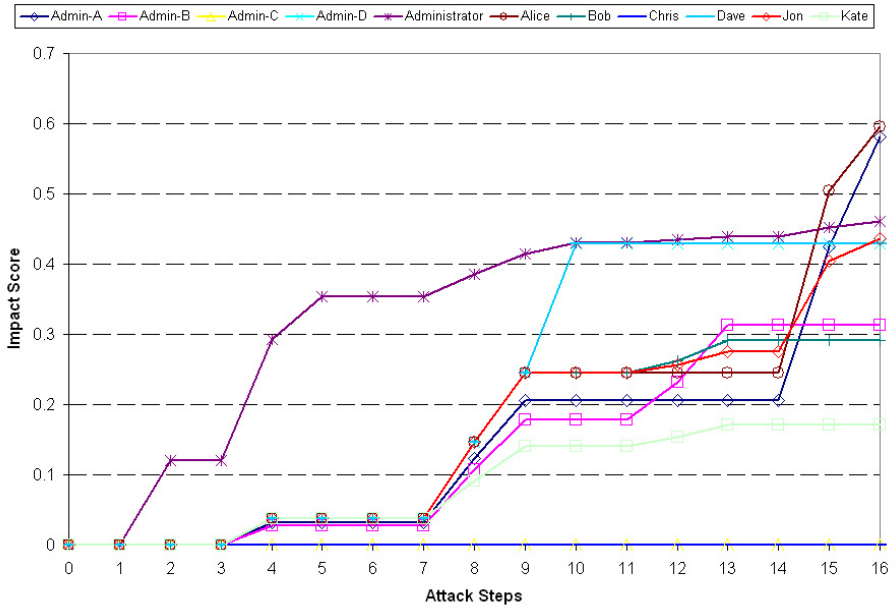

Figure 5.10: User impact scores for scenario 2

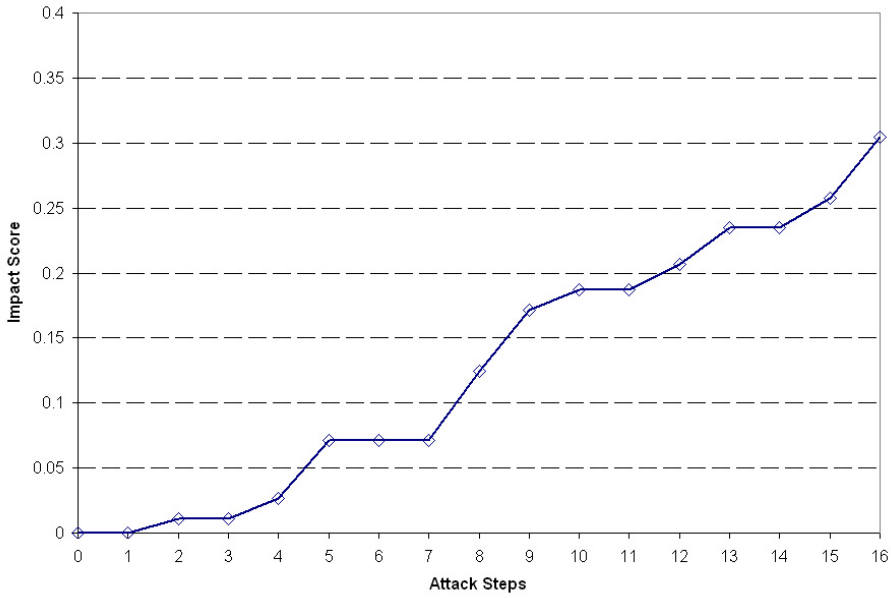

Figure 5.11: Network impact scores for scenario 2

5.2.2 Scenario 3

Figure 5.12 shows Scenario 3's attack progression. It is another fairly small attack, similar to Scenario 2, however it targets different machines. It starts by attacking the mail and web external servers, followed by the web, file, and SQL internal servers. From there, it attempts to attack Department C's cluster with a DoS attack. Refer to Table 5.7 for detailed attack steps.

Step	Description	Source IP	Dest IP	Port/Prot	Alert Signature
1	Scan External Web Server	30.54.126.213	192.168.1.2	80/tcp	WEB-MISC http directory traversal
2	Attack Ext. Web Server	30.54.126.213	192.168.1.2	80/tcp	WEB-IIS .asa HTTP header buffer overflow attempt
3	Attack Mail Server (POP3)	192.168.1.2	192.168.1.3	110/tcp	POP3 USER overflow attempt
4	Attack Mail Server (SMTP)	192.168.1.2	192.168.1.3	25/tcp	SMTP RCPT TO overflow
5	Attack Mail Server (SMTP)	192.168.1.2	192.168.1.3	25/tcp	SMTP exchange mime DOS
6	Compromise Internal Web Server	192.168.1.2	192.168.3.3	80/tcp	WEB-IIS .asa HTTP header buffer overflow attempt
7	Ping SQL from Int. Web Server	192.168.3.3	192.168.3.5	1433/tcp	MS-SQL ping attempt
8	Attack SQL from Int. Web Server	192.168.3.3	192.168.3.5	1433/tcp	WEB-MISC PCCS mysql database admin tool access
9	Attack Main File Server	192.168.3.3	192.168.3.4	445/tcp	NETBIOS nimda RICHED20.DLL
10	Attack Main File Server	192.168.3.3	192.168.3.4	445/tcp	NETBIOS SMB trans2open buffer overflow attempt
11	Attack Main File Server	192.168.3.3	192.168.3.4	445/tcp	NETBIOS DOS RFPoison
12	Scan Dep-C	192.168.3.3	192.168.31.100	456/icmp	ICMP PING Microsoft Windows
13	Attack Real Audio on Dep-C	30.54.126.213	192.168.31.100	276/tcp	DOS Real Audio Server

Table 5.7: Scenario 3's attack steps

Overall, it would make sense that a network administrator's I_U increase on almost all

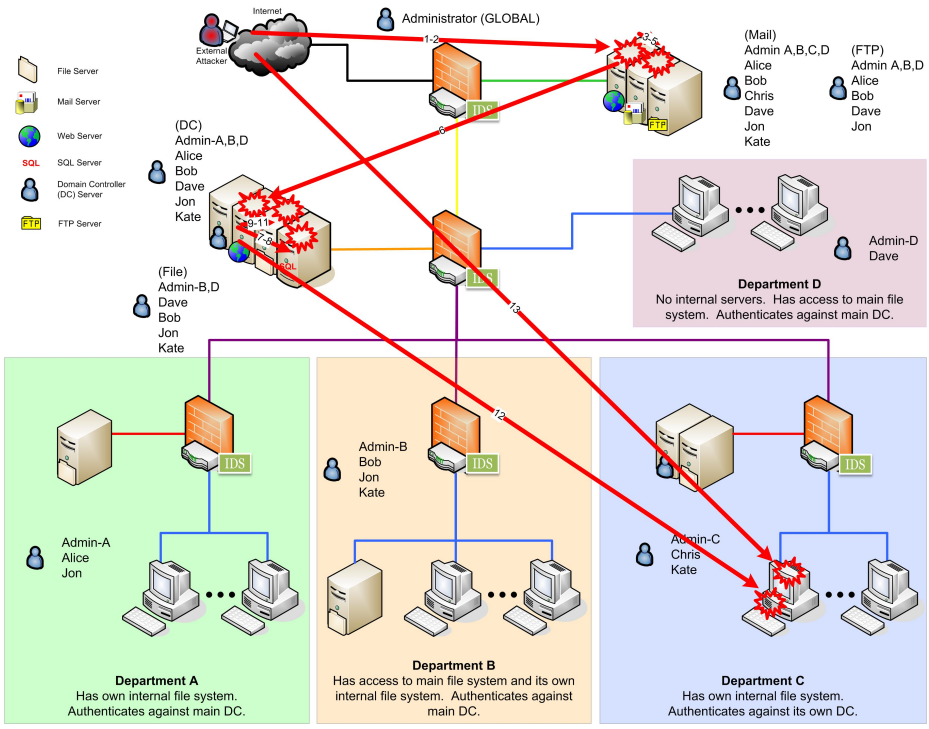

Figure 5.12: Topological view of scenario 3's attack steps

attack steps, because they oversee the entire network. However, there are two trends from scenario 3 that welcome discussion. In Figure 5.15, notice that Administrator is initially affected before any other user (this was also the case in scenario 2), and that their I_U is significantly higher than the other users'. The obvious answer to why the Administrator's I_U initially jumps before the other users' is because the other users do not have accounts on the machines that were attacked. This may be true, however another reason could be that the affected accounts may not be very important to the other users. The first few machines attacked are servers on the main external and internal subnets. When an outside attacker attacks the external servers and even the internal servers, it would appear that the global Administrator's I_U jumps fairly quickly. Therefore, the Administrator's I_U may be used as an early indication someone is attacking the external and internal server domains.

Using Figure 5.16, this scenario can be used to show the value of I_N-ME_S. The I_N-ME_S can be used to summarize how damaged the network potentially could be if the attacker fully exploited the services that they have already attacked. The interesting notion

about I_N-ME_S is that it almost acts as a predictor for attacks on single machines. For example, let us analyze the I_N-ME_S for the attack on the main file server, specifically attack steps 9-11. According to the main file server's I_H in Figure 5.13, it appears as though step 9 of the attack moderately harmless, causing only a small change in the I_H. This is also verified from the small increase in I_N shown in Figure 5.16. However, the steep rise of the I_N-ME_S compared to the shallow one of the I_N shows that potentially, there can be much more damage done to the network. Steps 10 and 11 continue to attack the NetBIOS service on the main file server, and the I_N catches up to the I_N-ME_S. This is the case for many of smaller series of attacks. Notice that the I_N does not always fully catch up to the I_N-ME_S. This remaining area is damage that can still technically be done on the network. For a longer attack, this gap has the capability of becoming quite large if all services are not fully exploited.

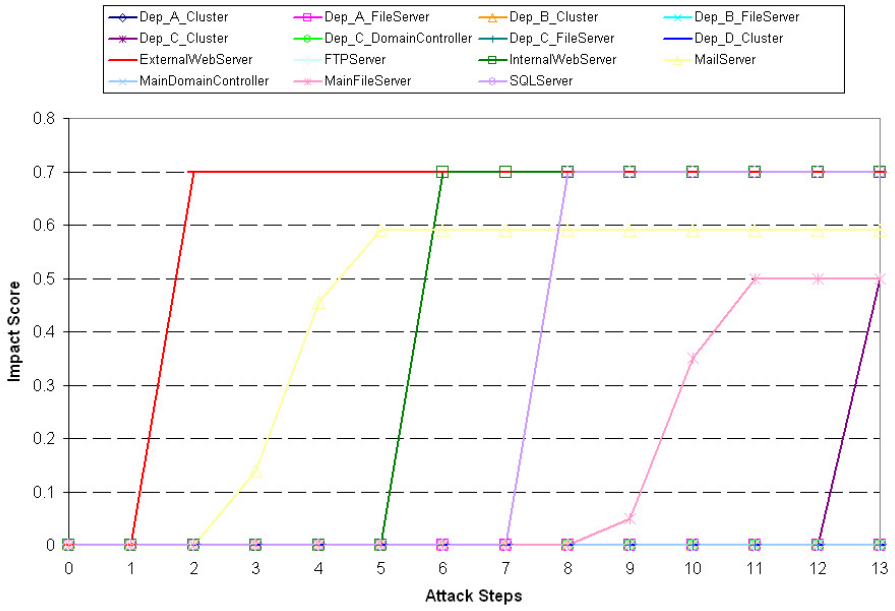

Figure 5.13: Host impact scores for scenario 3

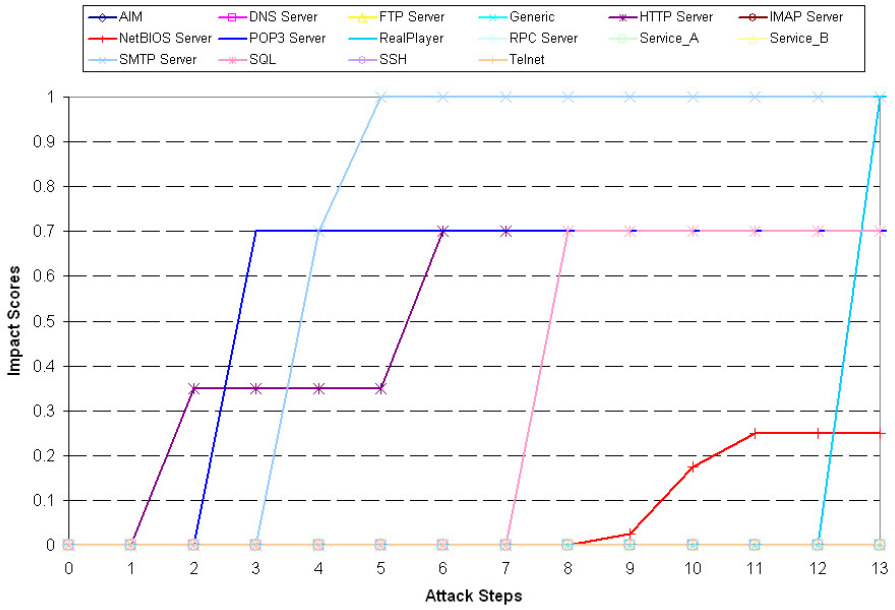

Figure 5.14: Service impact scores for scenario 3
5

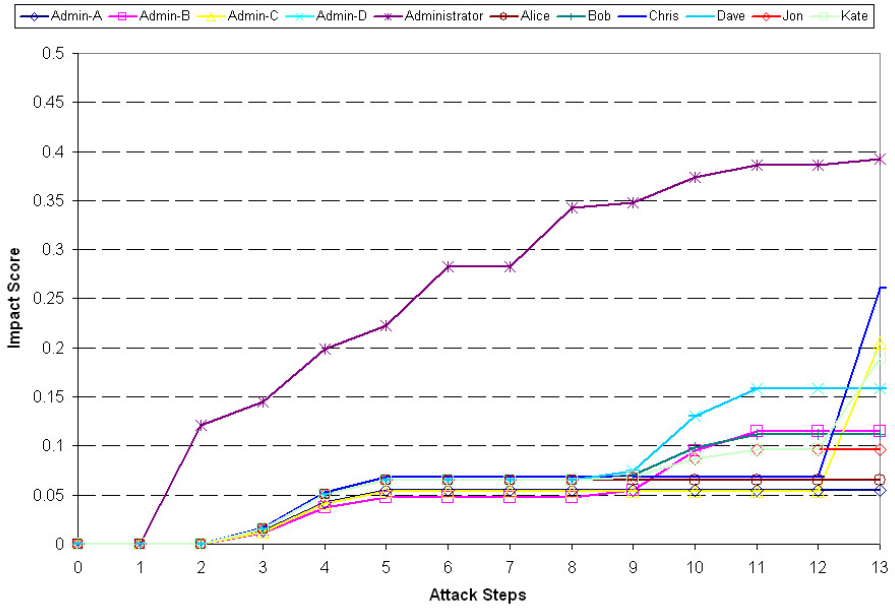

Figure 5.15: User impact scores for scenario 3

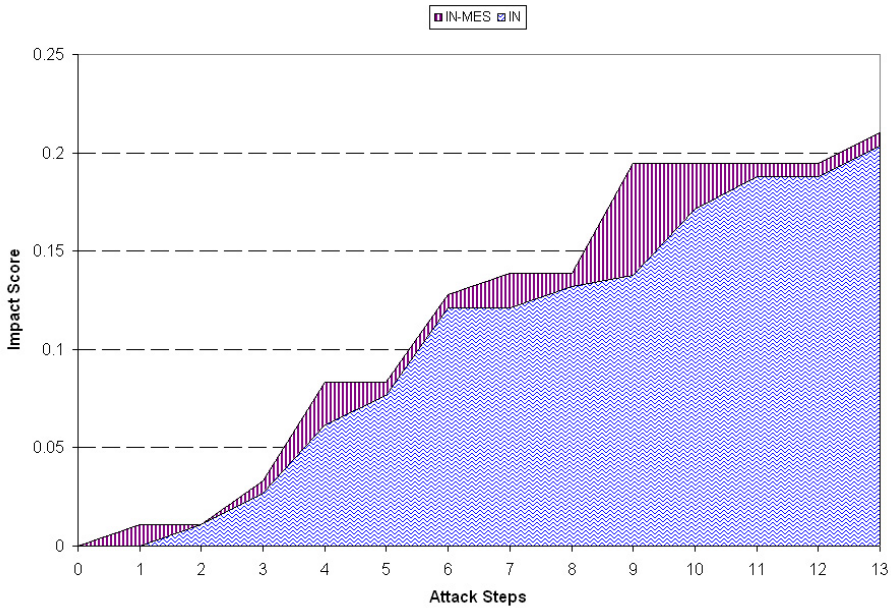

Figure 5.16: I_N and I_N-ME_S for scenario 3

5.2.3 Interweaving Attacks 2 & 3

This scenario involves analyzing what would happen if scenarios 2 and 3 were executed simultaneously, interweaving their attack steps (see Table 5.8). Since the two scenarios mostly target different machines, performing them together should results in higher impact scores for a larger variety of users. Figures 5.17-5.20 show the impact scores per attack step. These plots show similar trends as those from scenarios 2 and 3, and may be analyzed in the same way. Notice that there are a lot more components that are affected because of the combined attacks.

Figures 5.21-5.24 display the final impact scores for each scenario and the combined scenario for each of the components. The I_H results in Figure 5.21 show that by combining the attacks, the combined I_H appears to be the maximum of the I_H from scenarios 2 and 3. However, the max function is not quite valid. The problem is not in the scoring, the scenarios need to be analyzed a bit further. The combination rule appears to be the max function because there are no cases where multiple services are attacked on the same host

Attack	Step	Description	Source IP	Dest IP	Alert Signature
2	1	Scan External Web Server	140.203.195.48	192.168.1.2	WEB-MISC http directory traversal
2	2	Attack Ext. Web Server	140.203.195.48	192.168.1.2	WEB-IIS .asa HTTP header buffer overflow attempt
2	3	Scan FTP Server	192.168.1.2	192.168.1.4	FTP adm scan
2	4	Attack FTP Server	192.168.1.2	192.168.1.4	FTP ADMw0rm ftp login attempt
3	1	Scan External Web Server	30.54.126.213	192.168.1.2	WEB-MISC http directory traversal
3	2	Attack Ext. Web Server	30.54.126.213	192.168.1.2	WEB-IIS .asa HTTP header buffer overflow attempt
2	5	Compromise Internal Web Server	192.168.1.2	192.168.3.3	WEB-IIS .asa HTTP header buffer overflow attempt
3	3	Attack Mail Server (POP3)	192.168.1.2	192.168.1.3	POP3 USER overflow attempt
3	4	Attack Mail Server (SMTP)	192.168.1.2	192.168.1.3	SMTP RCPT TO overflow
3	5	Attack Mail Server (SMTP)	192.168.1.2	192.168.1.3	SMTP exchange mime DOS
2	6	Ping Dep-D	192.168.3.3	192.168.4.100	ICMP PING Microsoft Windows
2	7	Get info about Main DC	192.168.3.3	192.168.3.2	DNS named version attempt
2	8	Attack Main DC	192.168.3.3	192.168.3.2	DNS EXPLOIT named overflow attempt
2	9	Attack Main DC Again	192.168.3.3	192.168.3.2	RPC status GHBN format string attack
2	10	Apply Backdoor to DepD	192.168.3.3	192.168.4.100	BACKDOOR subseven DEFCON8 2.1 access
3	6	Compromise Internal Web Server	192.168.1.2	192.168.3.3	WEB-IIS .asa HTTP header buffer overflow attempt
2	11	Scan DepB File Server	192.168.3.3	192.168.20.2	NETBIOS SMB Startup Folder access attempt
2	12	Gain access to DepB File Server	192.168.3.3	192.168.20.2	NETBIOS SMB C$ access
3	7	Ping SQL	192.168.3.3	192.168.3.5	MS-SQL ping attempt
3	8	Attack SQL	192.168.3.3	192.168.3.5	WEB-MISC PCCS mysql database admin tool access
2	13	Attack DepB File Server	192.168.3.3	192.168.20.2	NETBIOS DOS RFPoison
3	9	Attack Main File Server	192.168.3.3	192.168.3.4	NETBIOS nimda RICHED20.DLL
3	10	Attack Main File Server	192.168.3.3	192.168.3.4	NETBIOS SMB trans2open buffer overflow attempt
2	14	Scan for SSH on DepA Cluster	192.168.20.2	192.168.11.100	SCAN SSH Version map attempt
3	11	Attack Main File Server	192.168.3.3	192.168.3.4	NETBIOS DOS RFPoison
2	15	Attack Dep-A Cluster SSH	192.168.20.2	192.168.11.100	EXPLOIT ssh CRC32 overflow
3	12	Scan Dep-D	192.168.3.3	192.168.31.100	ICMP PING Microsoft Windows
2	16	Attack DepA File Server	192.168.11.100	192.168.10.2	NETBIOS DOS RFPoison
3	13	Attack Real Audio on Dep-D	30.54.126.213	192.168.31.100	DOS Real Audio Server

Table 5.8: Interweaving attack steps of scenarios 2 and 3

by a separate attack, thus one of the scenario's impact scores is 0. Therefore, referring back to Figure 3.2, the I_H score will not change if other services are not asserted. The lack of services is due to the limited test network being used.

Similar to the combined I_H results, most of the combined I_S results in Figure 5.22 appear to be the result of a max function. However, the same idea applies here for the services as it did for the hosts. The NetBIOS service is attacked by scenario 2 on steps 11, 12, 13, and 16 and by scenario 3 on steps 9, 10, and 11, therefore each scenario produces some kind of impact on the NetBIOS service. The combination rules for the I_S Algorithm (Figure 3.3) are used to determine the combined result.

The combined I_U results best contribute to how components impact scores may be largely affected if multiple attacks compromise different parts of the network. Figure 5.23 shows most of the combined scores being greater in value than each of the individual attacks. However, these combined scores depend on a variety of things such as: if same

exploits were made on the same machines and how important the accounts are to a user whose hosts attacked. Kate for example, has an I_U of 0.1716 after scenario 2 and 0.1890 from scenario 3. Combined, Kate has an I_U of 0.3606, which is basically the sum of the two individual scores. However Administrator has a I_U of 0.4604 from scenario 2, 0.3920 from scenario 3, and a combined score of 0.6713. The combined score is obviously not the sum the two individual scores.

Finally, Figure 5.24 shows the comparison of overall network scores. All of these results and plots can be used to show what the damage could be if multiple attacks are executed. They also may provide some insight on which attack affected a specific component the most.

This type of analysis may also be helpful for detecting coordinated attacks. For example, if one attack has illogical steps involved in it, then by combining other attacks in the attack analysis, more components are most likely going to be asserted. In a real-time system with automatic updates of network configurations, including ports opened by back door attacks, or changed firewall rules, these added asserted components may cause illogical attacks to disappear, implying a coordinated attack.

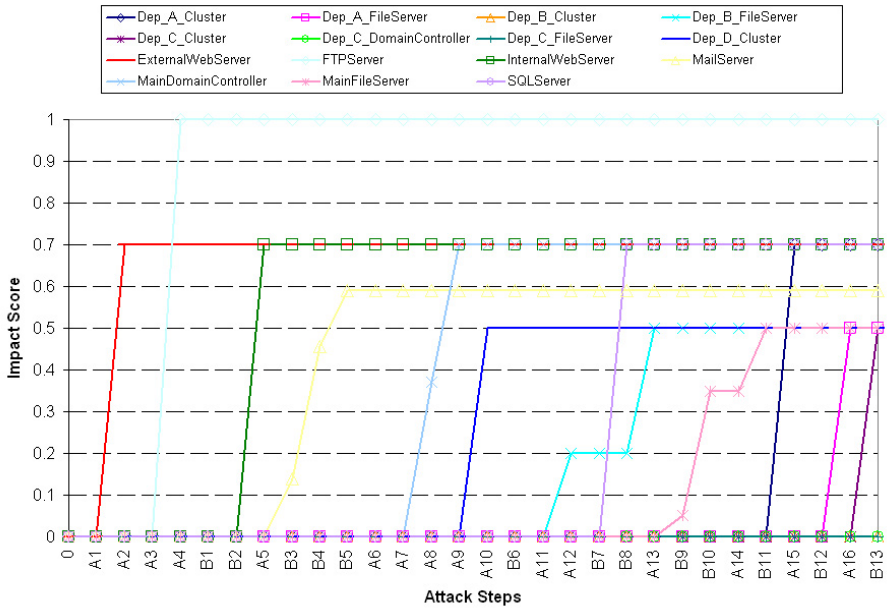

Figure 5.17: Host impact scores for interweaving scenarios 2 and 3

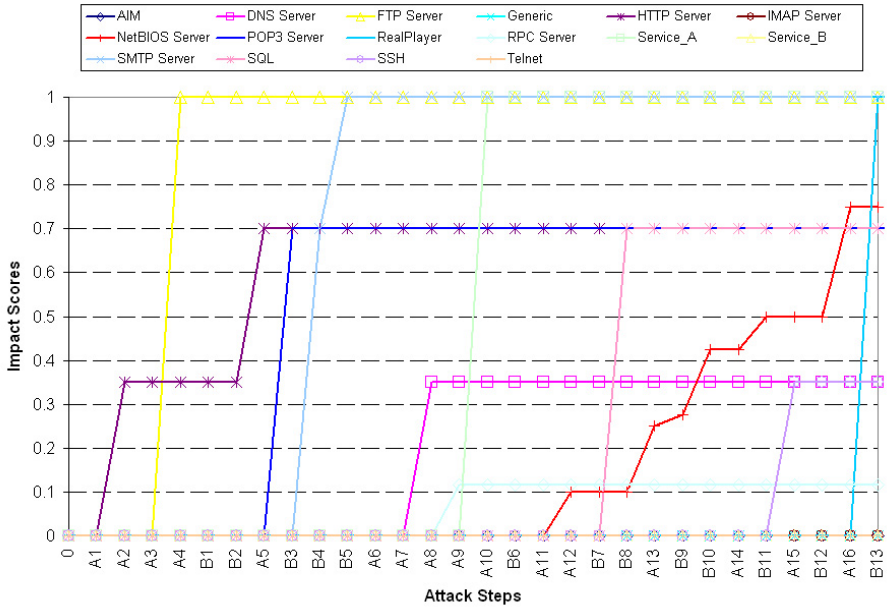

Figure 5.18: Service impact scores for interweaving scenarios 2 and 3

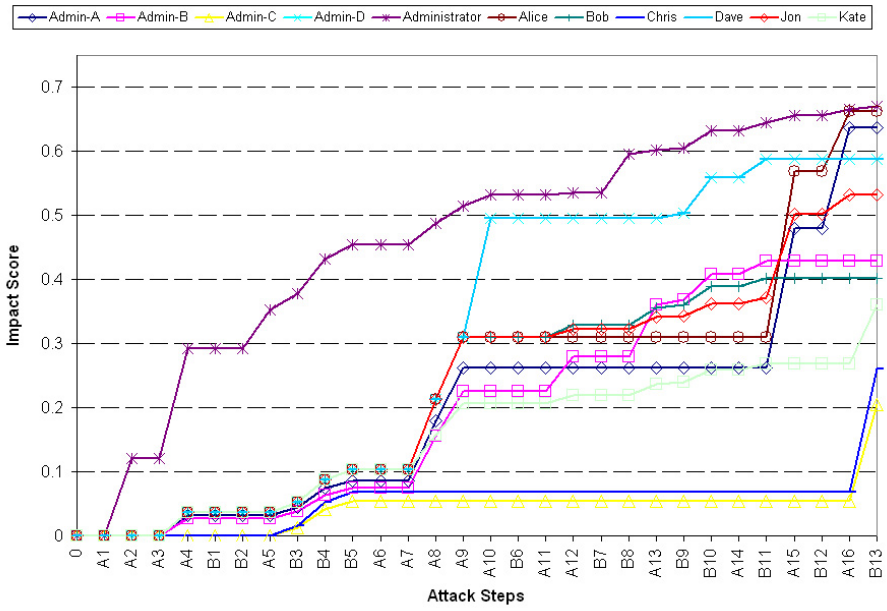

Figure 5.19: User impact scores for interweaving scenarios 2 and 3

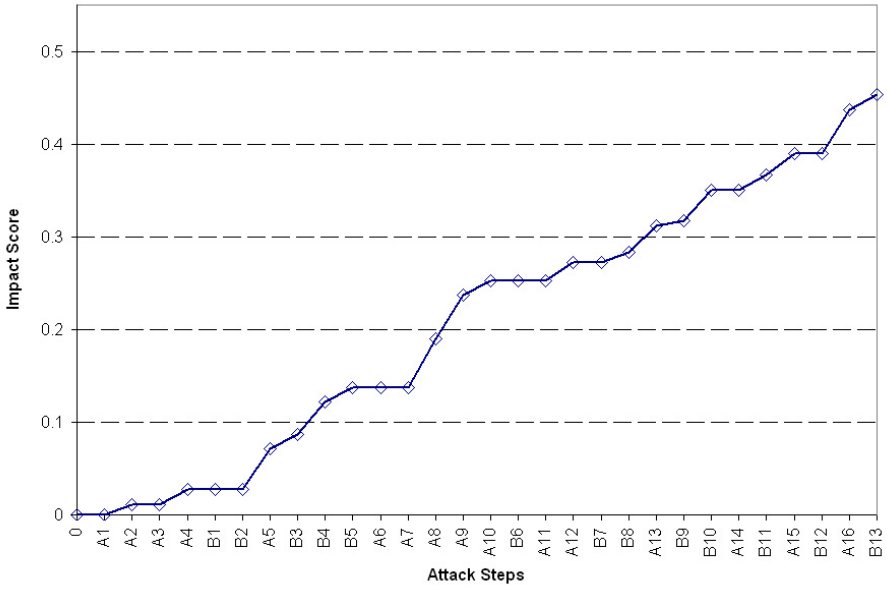

Figure 5.20: Network impact scores for interweaving scenarios 2 and 3

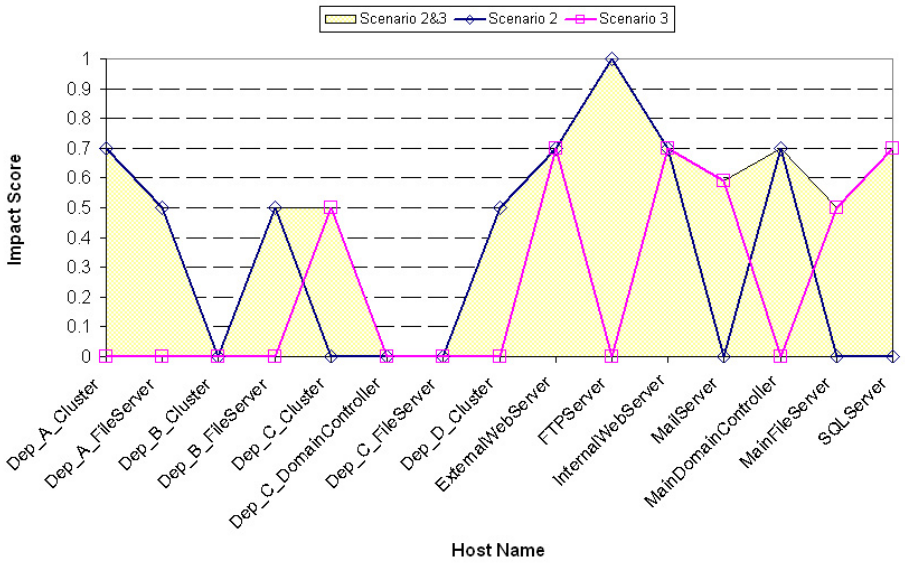

Figure 5.21: Final host impact scores for comparing scenarios 2, 3, and 2 & 3 together

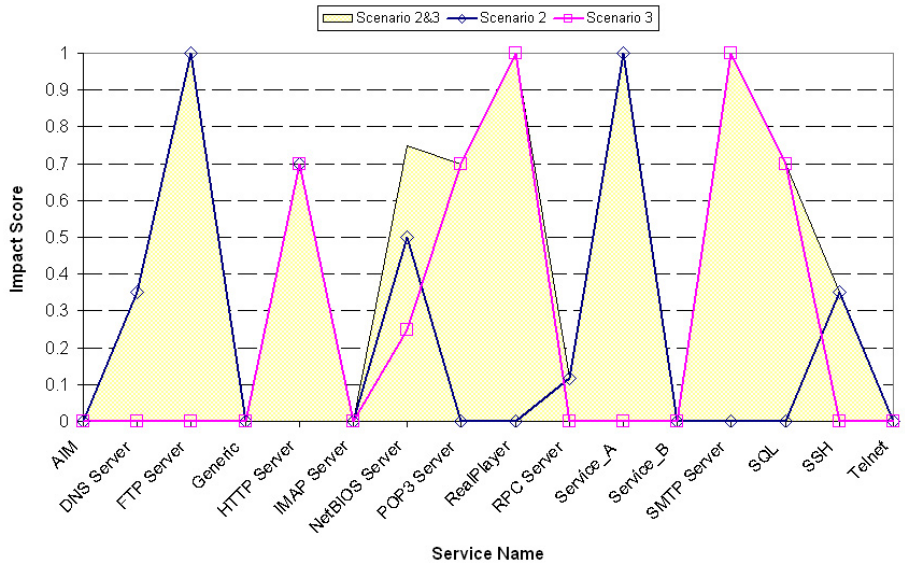

Figure 5.22: Final service impact scores for comparing scenarios 2, 3, and 2 & 3 together

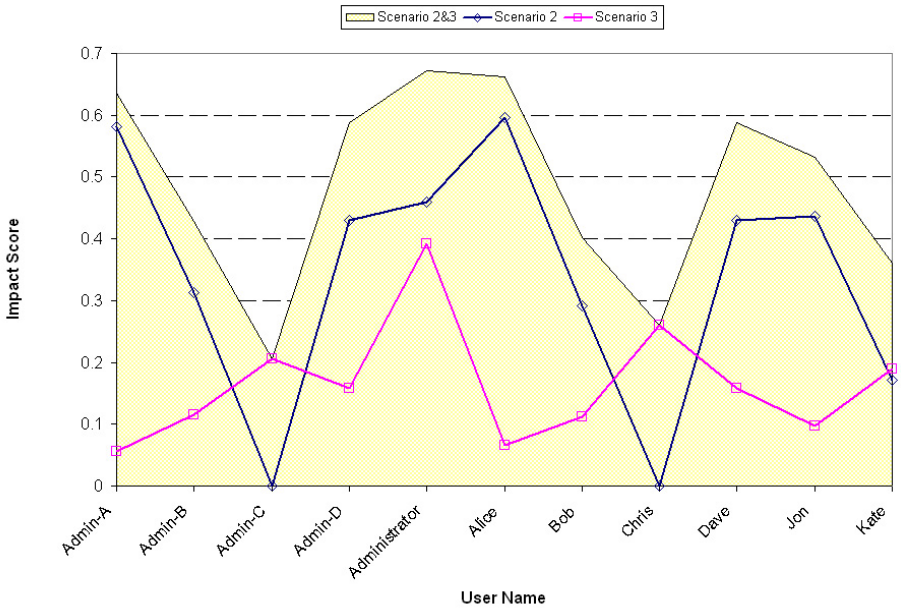

Figure 5.23: Final user impact scores for comparing scenarios 2, 3, and 2 & 3 together

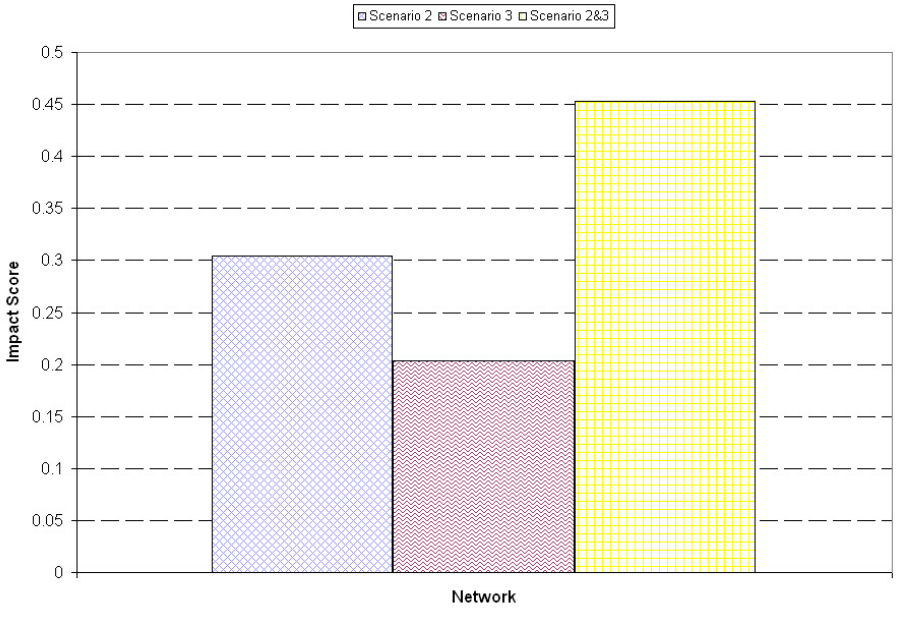

Figure 5.24: Final network impact scores for comparing scenarios 2, 3, and 2 & 3 together

5.2.4 Scenario 4 - Insider Attack

So far, only attack scenarios from outside threats have been analyzed. In today's world, insider threats are also common. These attacks may come from disgruntled employees, employees coerced into helping a hacker, a hacker physically gaining access to an inside machine, or possibly from an employee snooping around or showing off their hacking capabilities. Figure 5.25 show's an attack progression of an insider attack, initiated from the Department C cluster.

Consider the results shown in Figures 5.26-5.29 from the perspective of a network analyst. Noticing any major differences between an outsider attack and this one may lead to tell-tale signs that an insider attack has occurred. A few observations include:

- The first few affected hosts are not from the internal or external server domains.
- The services initially attacked to not match with any of the services on the external server domain.
- The global Administrator's I_U does not spike like it did in the previous scenarios.
- I_U for users with accounts in Department-C initially spike, particularly Admin-C's.

Another item that may flag concern is the fact that Department C is supposedly very secure. The network is configured such that anyone outside that department is not supposed to have access, especially to the servers.

Detecting insider attacks can be a difficult task and cause maximum damage to a network if combined with a second attack. For example, Kate has access to both Department B and C, giving her accounts on Department C's file server, DC, and cluster, along with Department B's file server, cluster, and the main file server and DC. Kate could potentially compromise all of those servers from an "already has access" point of view. Once she gathers the secure information such as user names and passwords of other people, she can either use the information herself or relay it to an outsider which would ease their attack.

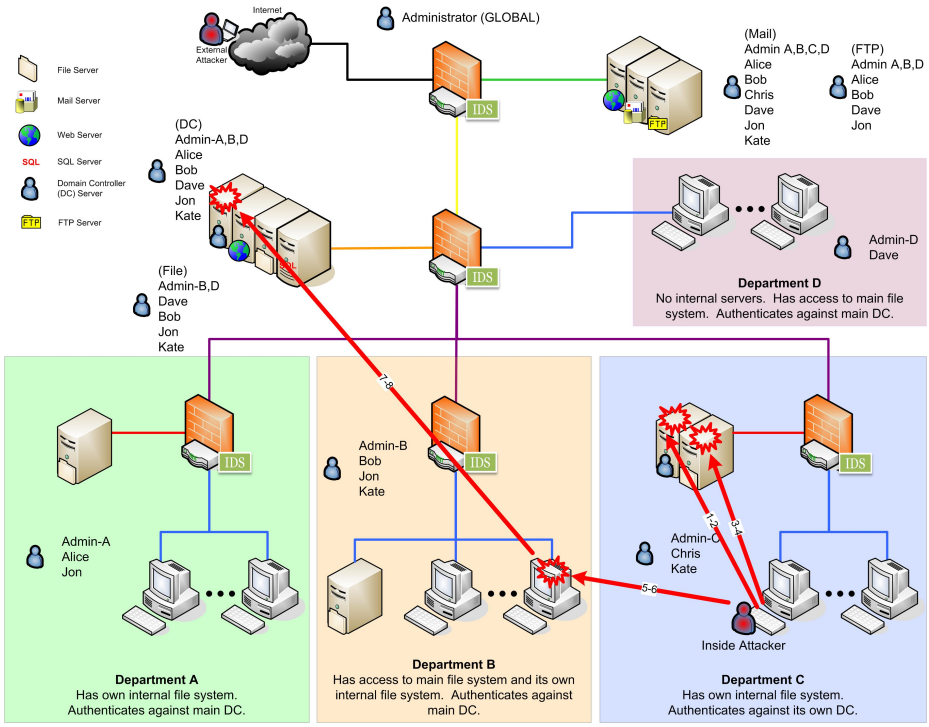

Figure 5.25: Topological view of scenario 4's attack steps

Step	Description	Source IP	Dest IP	Port/Prot	Alert Signature
1	Attack Dep-C DC	192.168.31.104	192.168.30.2	53/tcp	DNS named version attempt
2	Attack Dep-C DC	192.168.31.104	192.168.30.2	53/tcp	DNS EXPLOIT named overflow attempt
3	Attack Dep-C File Server	192.168.31.104	192.168.30.3	139/tcp	NETBIOS SMB C$ access
4	Attack Dep-C File Server	192.168.31.104	192.168.30.3	139/tcp	NETBIOS SMB ADMIN$access
5	Attack Dep-B Cluster	192.168.31.104	192.168.20.102	22/tcp	SCAN SSH Version map attempt
6	Attack Dep-B Cluster	192.168.31.104	192.168.20.102	22/tcp	ATTACK-RESPONSE success gobbles ssh explt (GOBBLE)
7	Attack Main DC	192.168.20.102	192.168.3.2	135/tcp	RPC mountd TCP exportall request
8	Attack Main DC	192.168.20.102	192.168.3.2	53/tcp	DNS EXPLOIT named 8.2-to-8.21

Table 5.9: Scenario 4's attack steps

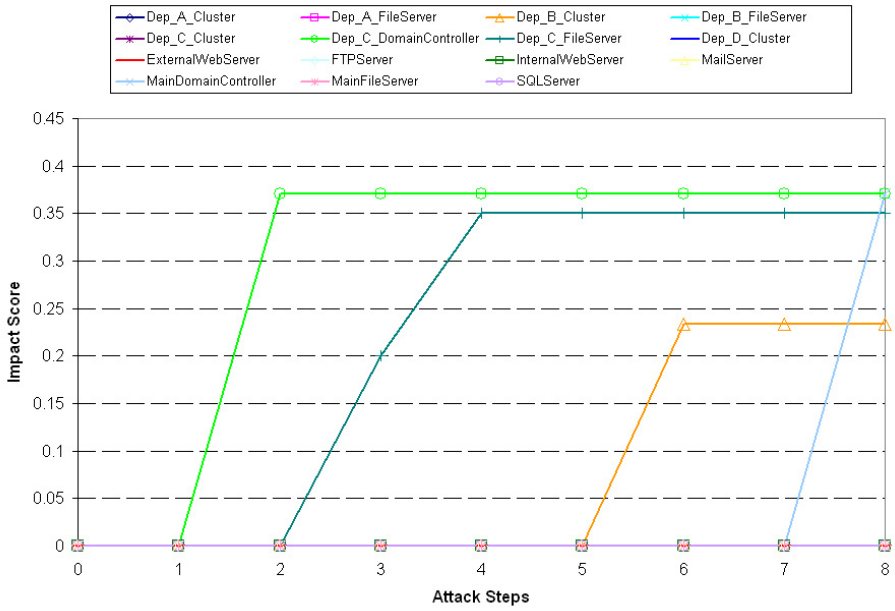

Figure 5.26: Host impact scores for scenario 4

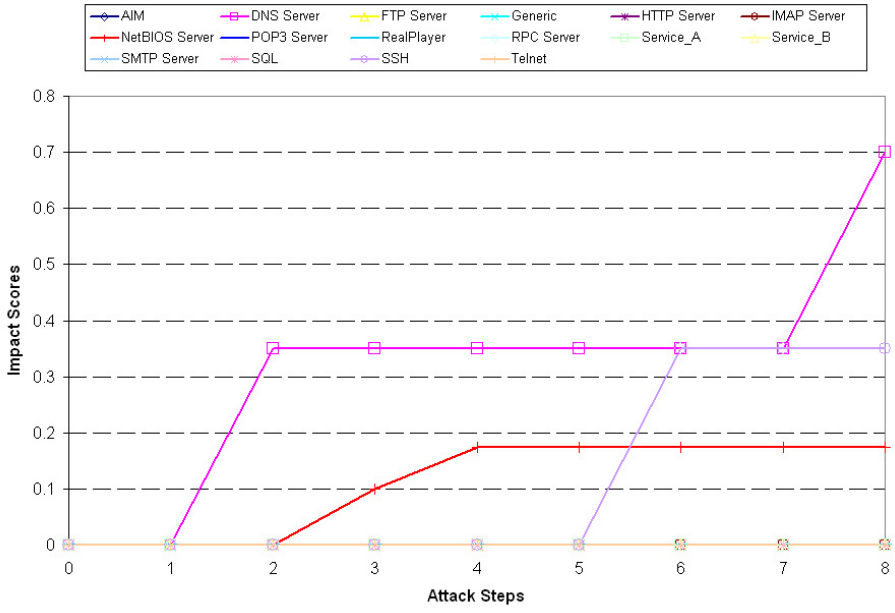

Figure 5.27: Service impact scores for scenario 4

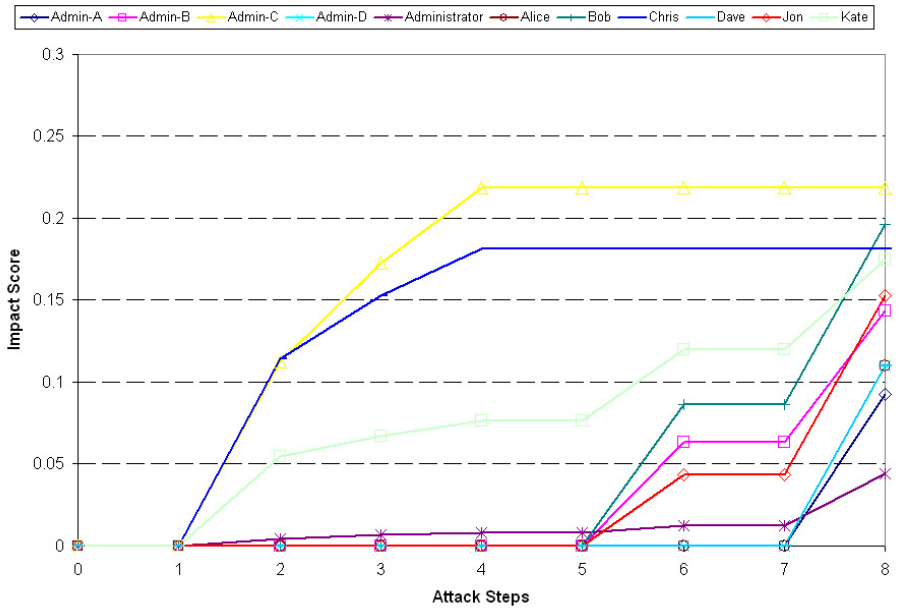

Figure 5.28: User impact scores for scenario 4

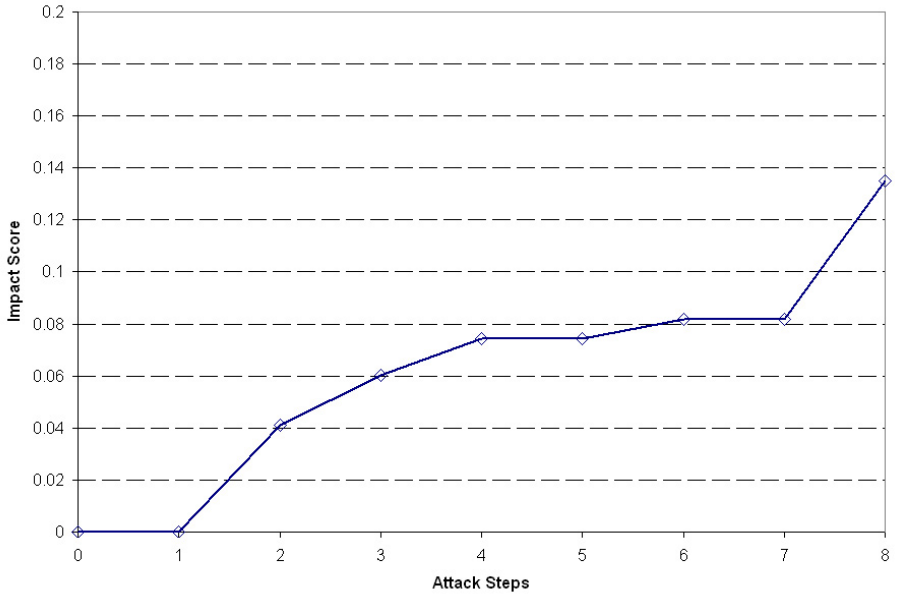

Figure 5.29: Network impact scores for scenario 4

5.2.5 Scenario 5 - Logical vs. Illogical Attack Steps

Scenario 5 was created to experiment with illogical attacks steps. This scenario was run twice, first not allowing illogical attacks to be processed, and then a second time allowing them to be. The scenario shown in Figure 5.30 has four illogical attack steps, steps 9-12 (see Table 5.10). Once the hacker gains access to the internal server domain, they attempt

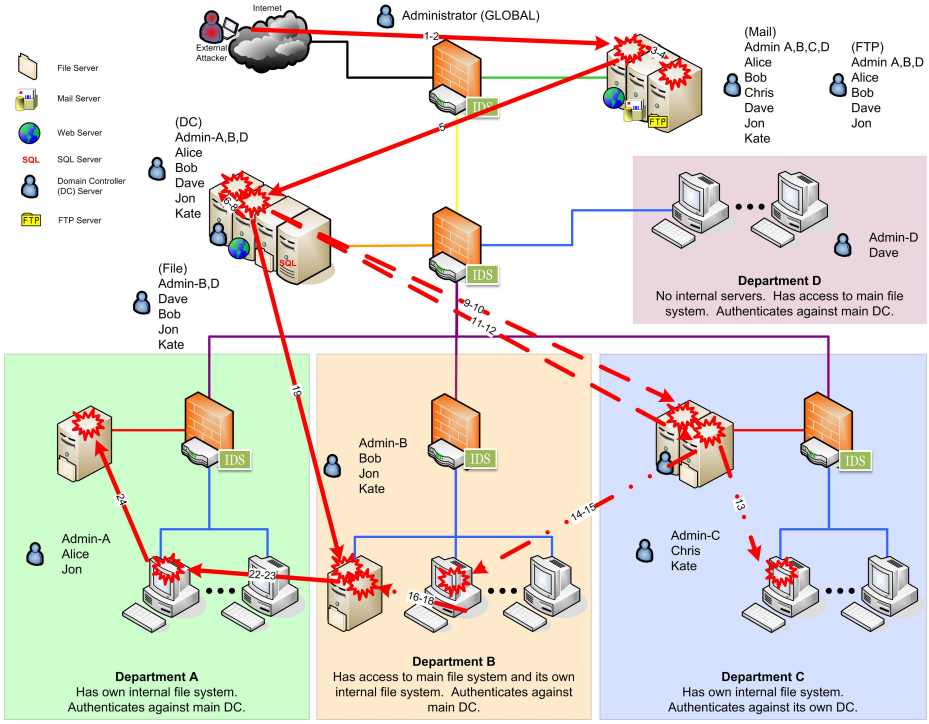

Figure 5.30: Topological view of scenario 5's attack steps [1]

to attack Department C's servers. The configuration of the network does not allow these servers to be accessed from outside of the department, thus they are labeled illogical. After this, the Department C servers become an illegal stepping stone to propagate further into the network, and attacks are executed from there. From these servers, there is a sequence of six steps that are considered valid, however the source of the attack is from a machine that was compromised using illogical attacks. Following these attacks, the attacker returns

[1]The dashed attack lines are illogical attacks. The dotted-dashed attack lines are logical attacks from a machine that was compromised using illogical attacks.

to the internal web server and continues to exploit other areas of the network. For easy comparison, each of the I_X results for both the logical and illogical processed attacks are displayed on the same page. Refer to Figures 5.31-5.37 for the results.

Step	Description	Source IP	Dest IP	Port/Prot	Alert Signature
1	Scan External Web Server	140.203.195.48	192.168.1.2	80/tcp	WEB-MISC http directory traversal
2	Attack Ext. Web Server	140.203.195.48	192.168.1.2	80/tcp	WEB-IIS .asa HTTP header buffer overflow attempt
3	Scan FTP Server	192.168.1.2	192.168.1.4	21/tcp	FTP adm scan
4	Attack FTP Server	192.168.1.2	192.168.1.4	21/tcp	FTP ADMw0rm ftp login attempt
5	Compromise Internal Web Server	192.168.1.2	192.168.3.3	80/tcp	WEB-IIS .asa HTTP header buffer overflow attempt
6	Get info about Main DC	192.168.3.3	192.168.3.2	/	DNS named version attempt
7	Attack Main DC	192.168.3.3	192.168.3.2	/	DNS EXPLOIT named overflow attempt
8	Attack Main DC Again	192.168.3.3	192.168.3.2	/	RPC status GHBN format string attack
9*	Recon on Dep-C DC	192.168.3.3	192.168.30.2	/	DNS zone transfer TCP
10*	Attack Dep-C DC	192.168.3.3	192.168.30.2	/	DNS EXPLOIT named 8.2-to-8.21
11*	Access Dep-C File Server	192.168.3.3	192.168.30.3	/	NETBIOS SMB ADMIN$access
12*	Attack Dep-C File Server	192.168.3.3	192.168.30.3	/	NETBIOS DOS RFPoison
13	Attack Dep-C Cluster	192.168.30.3	192.168.31.100	/	DOS Real Audio Server
14	Attack Dep-B Cluster SSH	192.168.30.2	192.168.20.100	/	SCAN SSH Version map attempt
15	Attack Dep-B Cluster SSH	192.168.30.2	192.168.20.100	/	EXPLOIT ssh CRC32 overflow
16	Scan Dep-B File Server	192.168.20.100	192.168.20.2	/	NETBIOS SMB Startup Folder access attempt
17	Gain access to Dep-B File Server	192.168.20.100	192.168.20.2	/	NETBIOS SMB C$ access
18	Attack Dep-B File Server	192.168.20.100	192.168.20.2	/	NETBIOS DOS RFPoison
19	Attack Dep-B File Server	192.168.3.3	192.168.20.2	/	NETBIOS DOS RFPoison
20	Scan for SSH on Dep-A Cluster	192.168.20.2	192.168.11.100	/	SCAN SSH Version map attempt
21	Attack Dep-A Cluster SSH	192.168.20.2	192.168.11.100	/	EXPLOIT ssh CRC32 overflow
22	Attack Dep-A File Server	192.168.11.100	192.168.10.2	/	NETBIOS DOS RFPoison

Table 5.10: Scenario 5's attack steps. [2]

Notice in Figure 5.31 that there are no changes for the I_H of Department C's domain controller and file server. This verifies that the illogical attacks were not processed by the impact assessment engine. All of the plots showing the results from processing illogical attacks have no changes in the I_X for steps 9-12.

Processing illogical steps could increase the overall network impact (Figure 5.37) and the impact scores of individual host, service, or user components. If scores do increase, and then logical steps follow that affect the same components, these impact scores could become misleading. For example, notice in Figures 5.33 and 5.34 that the I_S for the DNS and NetBIOS services end up approximately 0.35 and 0.25 higher when illogical attacks are processed, when according to the terrain model these attacks can not happen. Users from Department C also incur inflated impact scores. Impact scores from processing illogical attacks may be another score that analysts can compare against. It provides the potential

[2] Step numbers with an * next to them are Illogical attacks.

impact if the attack was actually successful, even though the model says it can't be.

With regard to attacks whose source is a machine compromised from an illogical attack, having the illogical attack flag available may be useful to the analyst. When such scenarios occur, either the terrain is configured incorrectly, a valid IDS alert was mistakingly filtered out, or that a hacker may have created a back door or reconfigured firewall rules.

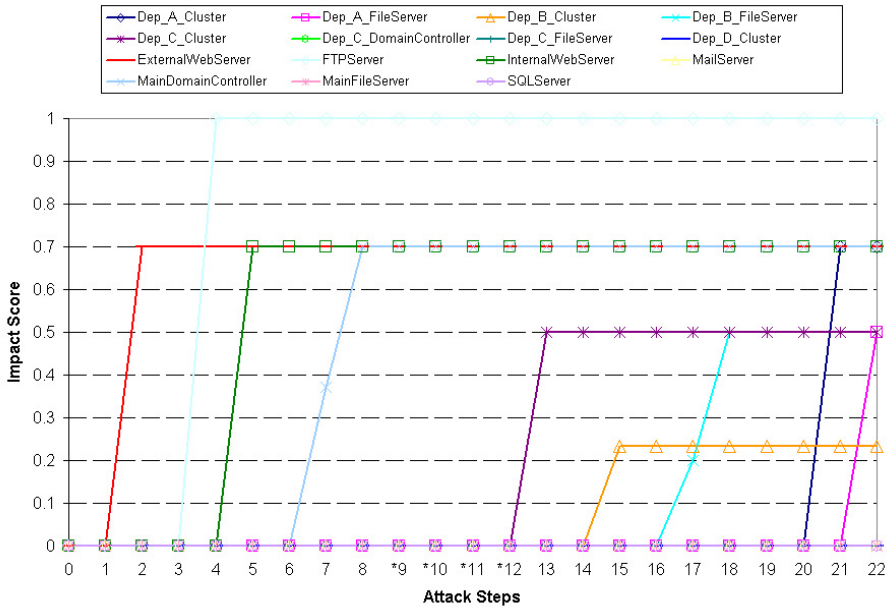

Figure 5.31: Host impact scores for scenario 5, logical attack processing

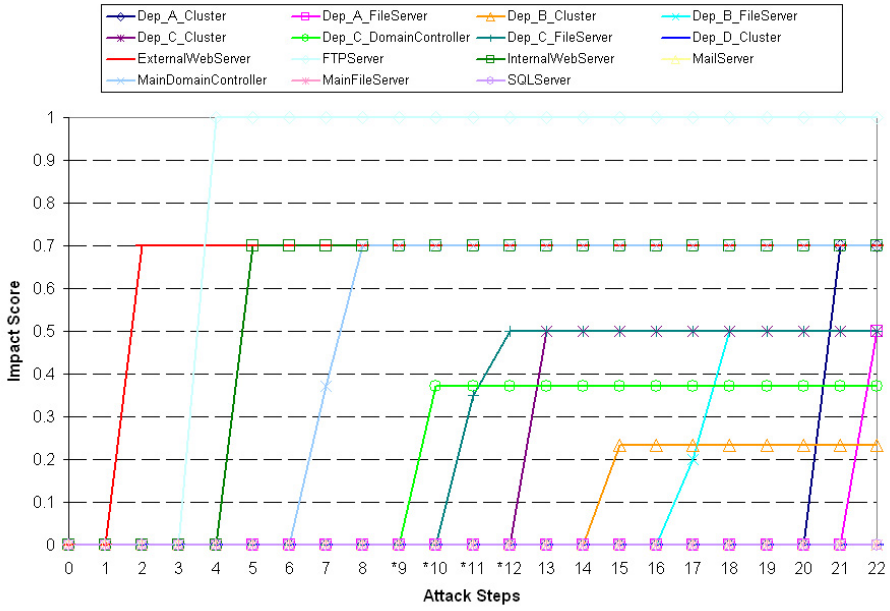

Figure 5.32: Host impact scores for scenario 5, illogical attack processing

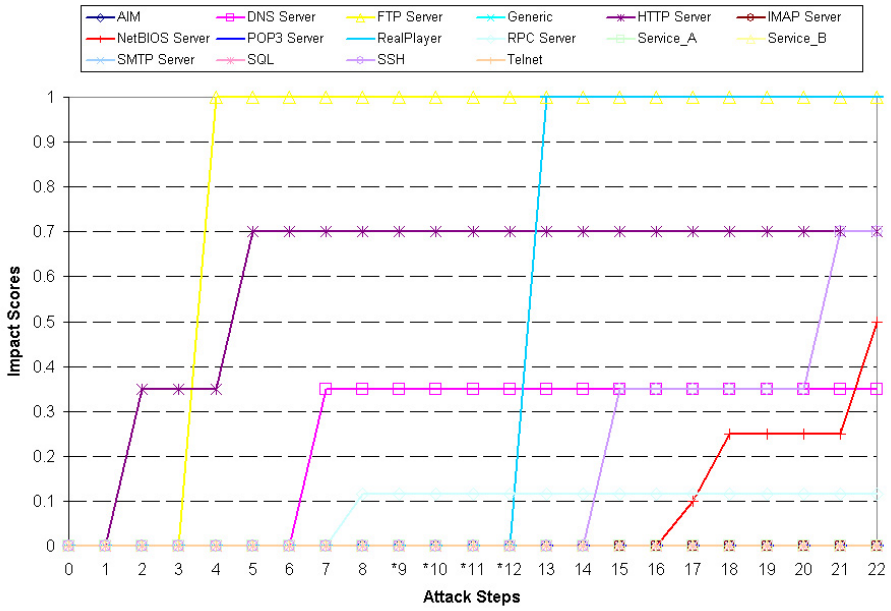

Figure 5.33: Service impact scores for scenario 5, logical attack processing

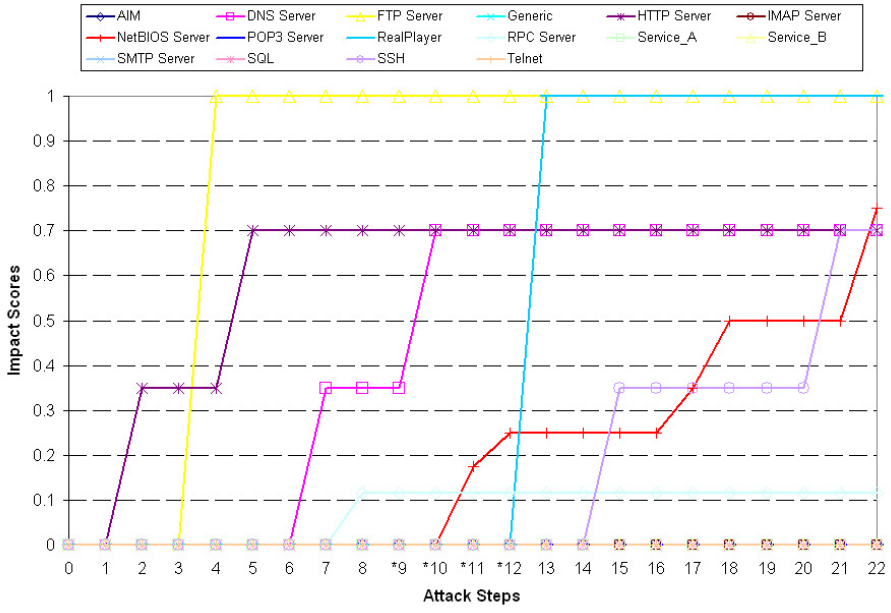

Figure 5.34: Service impact scores for scenario 5, illogical attack processing

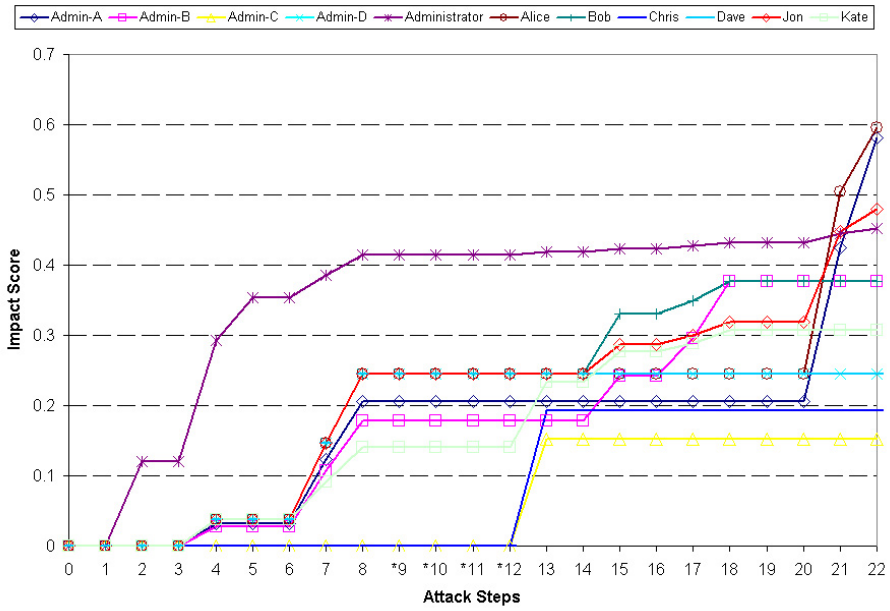

Figure 5.35: User impact scores for scenario 5, logical attack processing

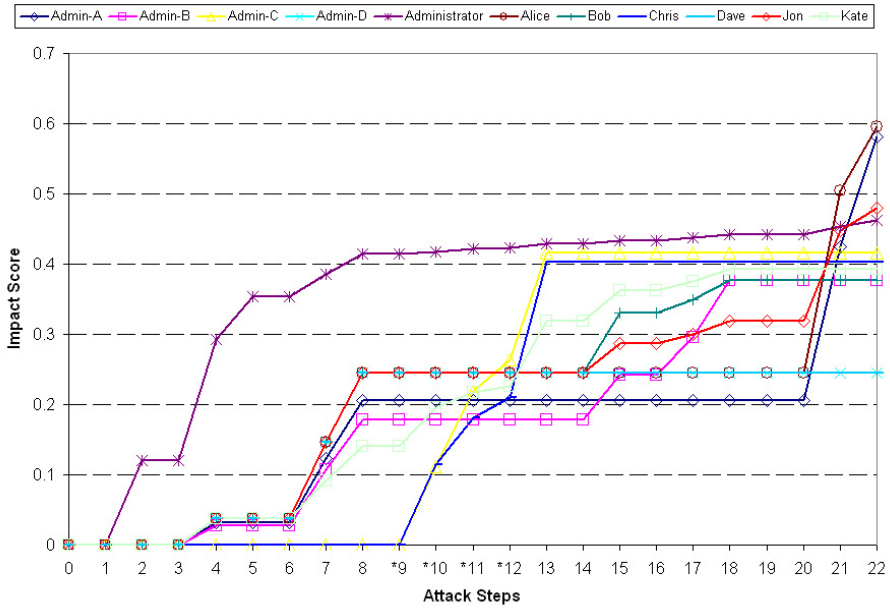

Figure 5.36: User impact scores for scenario 5, illogical attack processing

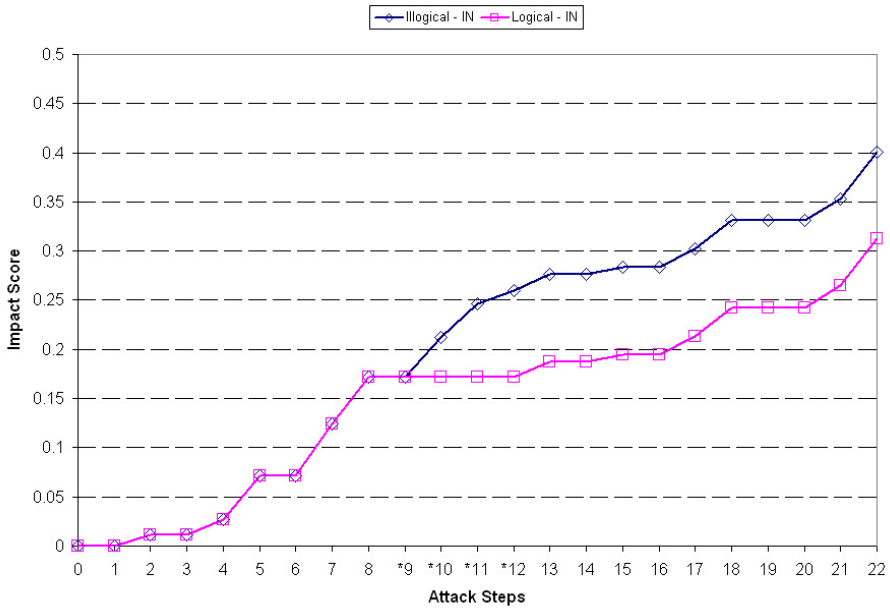

Figure 5.37: Network impact scores for scenario 5, combined view

5.2.6 Scenario 7 - Department Attacks

This scenario was specifically created to show how different configurations can affect impact scores. The same attack scenario (Figure 5.38) is run four times, each on a different department. Each of the four departments' cluster are setup with the same exact telnet service that will be exploited. I_N scores are collected for each scenario. In order for the I_N to be related to the department, the machine criticalities are set with respect to the department, not to the overall network. The four new schema configurations can be found on the thesis CD for more details.

Table 5.11 shows the attack steps with respect to Department A. Steps 1-14 are identical for all four department attack. Steps 15-17 change depending on what department is being attacked.

The department impact scores are shown in Figure 5.39. Departments A, B, and D all have very similar values and patterns, yet they are slightly different due to their different configurations. For example, Department A cannot access the main file server, however, Departments B and D can. When the main file server is attacked (steps 12-14), Department

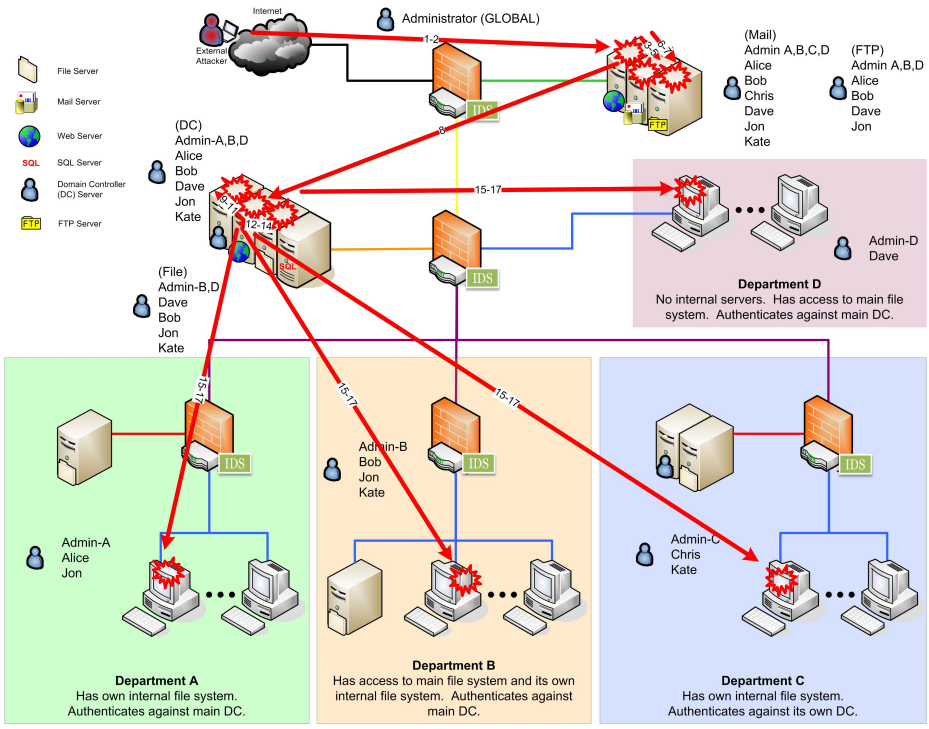

Figure 5.38: Topological view of scenario 7's attack steps

B and D's impact scores change, but Department A's does not. Department C appears to be least affected by the attack. This is because it uses its own authentication and file servers and does not have access to the FTP server. Therefore attacks on the FTP server, main file server, and main DC do not impact Department C. Also, although Department C has the telnet service on its cluster, the attack is illogical, because the router does not permit telnet traffic to pass through from the internal servers domain.

Table 5.12 shows the I_N-MEH for each department. Although they each have various configurations, it appears that their highest potential impact scores are all about the same.

Step	Description	Source IP	Dest IP	Port/Prot	Alert Signature
1	Scan External Web Server	140.203.195.48	192.168.1.2	80/tcp	WEB-MISC http directory traversal
2	Attack Ext. Web Server	140.203.195.48	192.168.1.2	80/tcp	WEB-IIS .asa HTTP header buffer overflow attempt
3	Attack Mail Server (POP3)	192.168.1.2	192.168.1.3	110/tcp	POP3 USER overflow attempt
4	Attack Mail Server (SMTP)	192.168.1.2	192.168.1.3	25/tcp	SMTP RCPT TO overflow
5	Attack Mail Server (SMTP)	192.168.1.2	192.168.1.3	25/tcp	SMTP exchange mime DOS
6	Scan FTP Server	192.168.1.2	192.168.1.4	21/tcp	FTP adm scan
7	Attack FTP Server	192.168.1.2	192.168.1.4	21/tcp	FTP ADMw0rm ftp login attempt
8	Compromise Internal Web Server	192.168.1.2	192.168.3.3	80/tcp	WEB-IIS .asa HTTP header buffer overflow attempt
9	Get info about Main DC	192.168.3.3	192.168.3.2	/	DNS named version attempt
10	Attack Main DC	192.168.3.3	192.168.3.2	/	DNS EXPLOIT named overflow attempt
11	Attack Main DC Again	192.168.3.3	192.168.3.2	/	RPC status GHBN format string attack
12	Attack Main File Server	192.168.3.3	192.168.3.4	445/tcp	NETBIOS nimda RICHED20.DLL
13	Attack Main File Server	192.168.3.3	192.168.3.4	445/tcp	NETBIOS SMB trans2open buffer overflow attempt
14	Attack Main File Server	192.168.3.3	192.168.3.4	445/tcp	NETBIOS DOS RFPoison
15	Attack Dep-A Cluster	192.168.3.3	192.168.11.102	23/tcp	WEB-MISC telnet attempt
16	Attack Dep-A Cluster	192.168.3.3	192.168.11.102	23/tcp	TELNET EZsetup account attempt
17	Attack Dep-A Cluster	192.168.3.3	192.168.11.102	23/tcp	TELNET bsd telnet exploit response

Table 5.11: Scenario 7's attack steps

Department	Impact Score
A	0.8542308
B	0.8538372
C	0.8503948
D	0.8542308

Table 5.12: I_N-ME_H for each department configuration

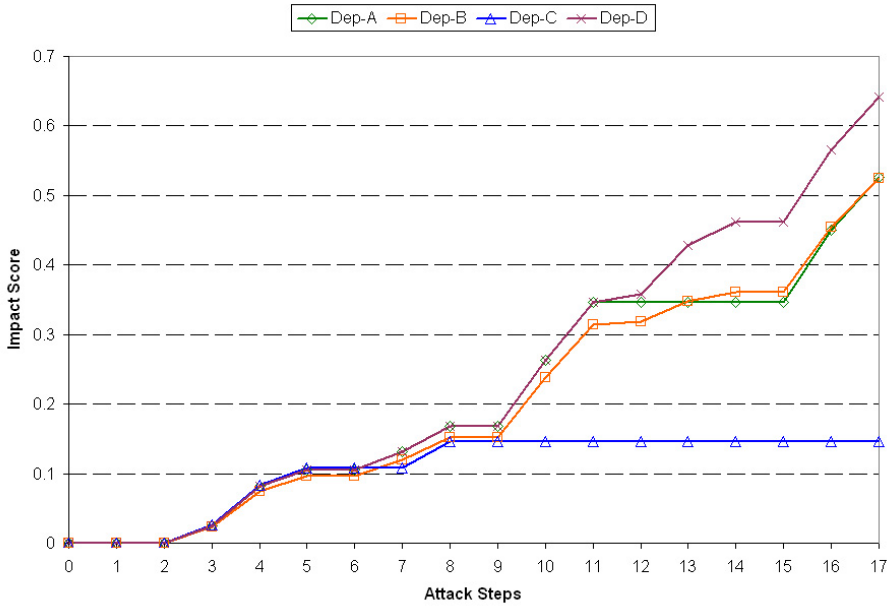

Figure 5.39: Individual department impact scores for scenario 7

5.2.7 Scenario 8 - Randomly Generated Attack Data

The previous scenarios were all manually created to help demonstrate different elements of the impact scores and how the overall system could potentially be used. Having attack data that is randomly generated can be used to solidify the capability of the system.

The steps in Tables 5.14-5.18 were randomly generated by an attack simulator developed by students in the RIT Industrial Engineering Department. Given a network that has machines defined with services and exposures, along with connectors (routers), the simulator has the capability to generate random attacks. The attacks are based on a variety of metrics including efficiency, stealth, skill, target, goal, step time, and start time. These metrics are primarily used to define how direct an attack path is, the tendency to raise alerts by achieving more goals, and to determine the success probability for each step. The simulator team created scenario with five separate attack tracks, one shown in each table. The metrics for the scenario are shown in Table 5.13.

Attack	Efficiency	Stealth	Skill	Target	Goal	Step Time	Start Time
First	0.3	1.0	0.999	192.168.3.4	Dos	5	0
Second	0.3	0.9	0.999	192.168.4.102	Backdoor	5	10
Third	0.4	0.95	0.999	192.168.20.101	Backdoor	4	14
Fourth	0.2	0.99	0.999	192.168.30.3	Dos	3	21
Fifth	0.5	1.0	0.999	192.168.31.103	Dos	5	26

Table 5.13: Attack parameters for simulator to generate Scenario 8

Currently, the simulator does not take into account router firewall rules, so it does not distinguish logical versus illogical attacks. These results assume that all of the generated attacks are logical.

The five individual attack tracks could be analyzed in two ways: together, showing the maximum damage occurred if the five attacks were coordinated, or separate. The first type of results and analysis are similar to that of scenarios 2, 3, and 2 & 3 combined. We have not yet analyzed a scenario with separate attack tracks. Figure 5.40 shows an example of this type of analysis. The plot shows Dave's user impact scores from each of the five separate attack tracks. For analysis purposes and because time information was not produced with the attack steps, it is assumed that the attacks steps occur simultaneously. Figure 5.40

demonstrates the changing impact scores and what an analyst might see as attack tracks progress. Initially, the analyst my be concerned about attack track 4. However, as time goes on it becomes clear that attack track 2 has the largest impact on Dave. This attack track analysis could be done for every component that has an impact score, giving the analyst insight on what attacks may be most harmful.

Step	Description	Source IP	Dest IP	Alert Signature
1	Recon Enumeration	84.31.173.66	192.168.1.2	WEB-MISC http directory traversal
2	Intrusion Root	112.124.121.1	192.168.1.3	SMTP sendmail 5.5.5 exploit
3	Escalation Service	143.9.55.104	192.168.1.3	SMTP RCPT TO overflow
4	Intrusion Root	192.168.1.3	192.168.3.5	MS-SQL/SMB sp adduser database user creation
5	Escalation Service	192.168.1.3	192.168.3.4	NETBIOS SMB trans2open buffer overflow attempt
6	Recon Enumeration	192.168.3.4	192.168.3.3	WEB-MISC http directory traversal
7	Goal Dos	192.168.3.4	192.168.3.4	NETBIOS DOS RFPoison

Table 5.14: Scenario 8's randomly generated Attack Track 1

Step	Description	Source IP	Dest IP	Alert Signature
1	Intrusion Root	145.115.200.103	192.168.1.3	SMTP sendmail 5.5.5 exploit
2	Escalation Service	226.129.132.52	192.168.1.3	POP3 USER overflow attempt
3	Intrusion Root	192.168.1.3	192.168.1.3	IMAP authenticate overflow attempt
4	Escalation Service	192.168.1.3	192.168.1.3	SMTP RCPT TO overflow
5	Escalation Service	192.168.1.3	192.168.1.3	SMTP RCPT TO overflow
6	Intrusion Root	192.168.1.3	192.168.1.3	IMAP authenticate overflow attempt
7	Intrusion Root	192.168.1.3	192.168.1.3	IMAP authenticate overflow attempt
8	Intrusion Root	192.168.1.3	192.168.1.3	IMAP authenticate overflow attempt
9	Intrusion Root	192.168.1.3	192.168.4.102	TELNET bsd telnet exploit response
10	Escalation Service	192.168.4.102	192.168.3.4	NETBIOS SMB trans2open buffer overflow attempt
11	Intrusion Root	192.168.3.4	192.168.3.5	MS-SQL/SMB sp adduser database user creation
12	Goal Backdoor	192.168.3.5	192.168.4.102	BACKDOOR subseven DEFCON8 2.1 access

Table 5.15: Scenario 8's randomly generated Attack Track 2

Step	Description	Source IP	Dest IP	Alert Signature
1	Recon Enumeration	125.235.227.212	192.168.1.2	WEB-MISC http directory traversal
2	Recon Enumeration	19.144.127.233	192.168.1.2	WEB-MISC http directory traversal
3	Escalation Service	93.91.218.181	192.168.1.3	POP3 USER overflow attempt
4	Intrusion Root	192.168.1.3	192.168.1.3	SMTP sendmail 5.5.5 exploit
5	Intrusion Root	192.168.1.3	192.168.1.3	IMAP authenticate overflow attempt
6	Recon Enumeration	192.168.1.3	192.168.1.2	WEB-MISC http directory traversal
7	Escalation Service	192.168.1.3	192.168.1.3	SMTP RCPT TO overflow
8	Intrusion Root	192.168.1.3	192.168.4.101	TELNET bsd telnet exploit response
9	Escalation Service	192.168.4.101	192.168.3.4	NETBIOS SMB trans2open buffer overflow attempt
10	Misc VirusTrojan	192.168.3.4	192.168.3.4	NETBIOS nimda RICHED20.DLL
11	Intrusion User	192.168.3.4	192.168.20.102	MISC AIM AddExternalApp attempt
12	Goal Backdoor	192.168.20.102	192.168.20.101	BACKDOOR HackAttack 1.20 Connect

Table 5.16: Scenario 8's randomly generated Attack Track 3

Step	Description	Source IP	Dest IP	Alert Signature
1	Intrusion Root	10.102.204.253	192.168.1.3	SMTP sendmail 5.5.5 exploit
2	Escalation Service	207.16.55.88	192.168.1.3	SMTP RCPT TO overflow
3	Intrusion Root	192.168.1.3	192.168.1.3	SMTP sendmail 5.5.5 exploit
4	Intrusion Root	192.168.1.3	192.168.1.3	SMTP sendmail 5.5.5 exploit
5	Recon Enumeration	192.168.1.3	192.168.4.105	WEB-MISC telnet attempt
6	Intrusion Root	192.168.1.3	192.168.4.105	TELNET bsd telnet exploit response
7	Escalation Service	192.168.4.105	192.168.3.4	NETBIOS SMB trans2open buffer overflow attempt
8	Intrusion User	192.168.3.4	192.168.3.2	DNS EXPLOIT named 8.2-to-8.21
9	Escalation Service	192.168.3.2	192.168.3.5	MS-SQL xp proxiedmetadata possible buffer overflow
10	Intrusion Root	192.168.3.5	192.168.4.104	WEB-MISC telnet attempt
11	Intrusion User	192.168.4.104	192.168.3.5	MS-SQL xp showcolv possible buffer overflow
12	Intrusion Root	192.168.3.5	192.168.3.5	MS-SQL/SMB sp adduser database user creation
13	Escalation Service	192.168.3.5	192.168.3.5	MS-SQL xp proxiedmetadata possible buffer overflow
14	Intrusion Root	192.168.3.5	192.168.30.3	RPC tooltalk UDP overflow attempt
15	Intrusion User	192.168.3.5	192.168.31.102	MISC AIM AddExternalApp attempt
16	Intrusion User	192.168.31.102	192.168.31.102	MISC AIM AddGame attempt
17	Escalation Service	192.168.31.102	192.168.30.2	DNS EXPLOIT named overflow attempt
18	Intrusion User	192.168.30.2	192.168.31.103	MISC AIM AddExternalApp attempt
19	Intrusion User	192.168.31.103	192.168.31.105	MISC AIM AddGame attempt
20	Goal Dos	192.168.31.105	192.168.30.3	NETBIOS SMB DCERPC ISystemActivator bind attempt

Table 5.17: Scenario 8's randomly generated Attack Track 4

Step	Description	Source IP	Dest IP	Alert Signature
1	Recon Enumeration	62.73.133.23	192.168.1.2	WEB-MISC http directory traversal
2	Intrusion Root	151.219.32.6	192.168.1.3	IMAP authenticate overflow attempt
3	Intrusion Root	192.168.1.3	192.168.1.3	IMAP authenticate overflow attempt
4	Intrusion User	192.168.1.3	192.168.4.100	TELNET EZsetup account attempt
5	Recon Enumeration	192.168.4.100	192.168.3.3	WEB-MISC http directory traversal
6	Intrusion Root	192.168.4.100	192.168.4.104	WEB-MISC telnet attempt
7	Recon Scanning	192.168.4.104	192.168.4.104	ICMP Traceroute
8	Recon Scanning	192.168.4.104	192.168.31.100	ICMP Traceroute
9	Goal Dos	192.168.4.104	192.168.31.103	DOS Real Audio Server

Table 5.18: Scenario 8's randomly generated Attack Track 5

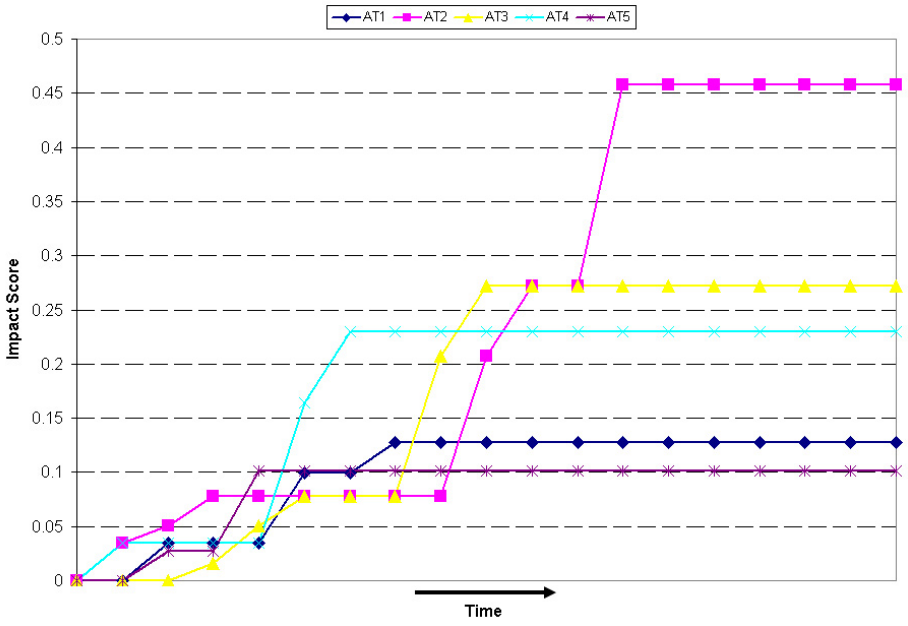

Figure 5.40: User Dave's impact score for each attack track in scenario 8

5.3 Limitations and Summary of Results

During the development of VTAC, some assumptions were made to make implementation more feasible:

1. Routers and switches are simply modeled as connectors. There is no information regarding services running on them, or considering them as an attack target.
2. The impact algorithms assume processing Router Neighbor Permission List rules with an Allowed attribute equal to true.
3. The impact algorithms assume processing Host Permission List rules with an Allowed attribute equal to false.
4. The impact algorithms do not consider the Privilege attributes in users, services, or exposures. The exposure damage score is used to embed the privilege information (discussed in Section 4.2.2).
5. Due to lack of publicly available fully defined networks, and scanning tools incapable of determining the overall mission of a network and its assets, arbitrary values used for criticality and damage scores.
6. Hosts can only have multiple IP addresses. They are limited to one LAN card.
7. Attack path calculations assume a spanning tree topology.
8. Local services are not accounted for.
9. Services are considered completely independent of each other. Valeur, *et al.*[29], propose a system where services may be dependent on one another.
10. There is no time limitation on how long an exposure may actually be affected. This prevents impact scores from decreasing over time.

Regardless of the limitations, the results presented in this chapter can provide an analyst with a detailed impact assessment of the current state of their network. The assessment heavily relies not only on a correct network configuration, but also numeric analysis. These numeric impact scores combined with the analyst's intuition and knowledge of the network, provide them with better situational awareness. It is vital to remember that an incorrect configuration or inaccurate criticalities can mislead the analyst's perception.

In order to properly create an accurate terrain model, a variety of scanners and administrator input is needed. As discussed in Chapter 2 Section 2.3, scanners with capabilities similar to Nessus or NMap should be used to gather machine, service, vulnerability, and connectivity data. Routers or switches need also be scanned to properly define permission rules between different subnets or domains. A list of users, and what machines they have accounts on is also required. A network administrator may have the inconvenience to define and update missing information such as criticality metrics for services, machines, and accounts. Ideally, all of this information could be automatically provided.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

The VTAC system has two main contributions. The first contribution was the development of a virtual cyber terrain used to model a network. The virtual terrain was designed to include network characteristics that we deemed necessary to perform impact assessment of cyber attacks. The terrain can be modeled as a directed graphs with attributed nodes and edges. A test network was created with diverse configurations to demonstrate the capability of the virtual terrain and the impact assessment algorithms. The second contribution was the development of impact assessment algorithms used for host, service, user, and network components. These algorithms make efficient use of the virtual terrain to gather information and update impact scores. The scores resulting from these algorithms represent the potential impact that an attack has had and can have on the component. A variety of scenarios were created and simulated to verify the virtual terrain configuration and the basic trends of the impact scores.

The development of this impact analysis system allows analysts to better understand and monitor the current state of their network. The scenario analysis provided insight on how noticing certain score trends could be used to help protect the network.

6.2 Future Work

Since the primary focus of this work was to introduce the concept of impact assessment using a network model, rather than implementing a full-featured working prototype, there are a number of possible extensions to this work. This section will discuss a few of these extensions and the reasoning behind them.

6.2.1 Different Attack Scenarios and Network Configurations

Although the results proved to be insightful, they were the result of a manually created test network and scenarios. Simulating the impact system over more realistic test networks with a variety of configurations could be used to help validate results. Additional randomly generated attack scenarios from the RIT simulator team, with the capability of producing all logical attack steps, or true attack data from a real network may also prove helpful to analyze the impact system.

6.2.2 Other Impact Algorithms

The designed impact algorithms were used to perform a rudimentary impact assessment using the virtual terrain model. A deeper analysis of combination rules and other impact scores is needed for future development. As mentioned in Section 5.3, privilege is not taken into account when determining the impact scores. Adding in privilege as a factor in the impact score may provide additional insight into the severeness of the attack.

6.2.3 Impact Projection

Section 3.2.6 discussed potential projection algorithms and how they can use of the virtual terrain. Having impact projection capabilities may enhance the analyst's situational awareness and allow them to make more informed decisions.

6.2.4 Real-Time Development

The current simulator presented in Section 4.1 does not factor in time of attack or exploit expiration in any way. Development of a real-time simulator could include injecting attacks at any time, rather than having a predefined list of steps executing at the push of a button. Also, cyber attacks often expire after a certain period of time. Including this information would make the impact system more realistic and indicative of critical attacks in real-time by observing the fluctuation in impact scores.

6.2.5 Integration Into Larger Defense System

Impact assessment provides only one of many critical pieces necessary for defending a network. Integration of VTAC into a larger defense system, where real attack tracks could be input to VTAC and other analysis algorithms could use the output of VTAC, could increase the overall situational awareness of the network and greatly aid a network analyst.

6.2.6 System Visualization

Visualization is an important aspect of complete systems. A system that has proper visualization can often add to the user's experience and the effectiveness of the software. VTAC's GUI was strictly developed for simulation purposes, not looks or effectiveness. There are a number enhancements that could be made to VTAC's appearance and capabilities. The virtual terrain could be able to be displayed as a network picture, similar to the figures used to display the attack scenarios. Clicking on an object could bring up all of the information associated with it. Impact scores could be integrated into the picture of the network, with scores above icons, or colors identifying impact severity. Real-time plots showing the impact scores for each component could also be readily available.

Bibliography

[1] Tim Bass. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4), April 2000.

[2] Microsoft Corporation. Microsoft security response center security bulletin severity rating system. `http://microsoft.com/technet/security/bulletin/rating.mspx`, 2002.

[3] F. Cuppens and A. Miege. Alert correlation in a cooperative intrusion detection framework. In *Proceedings 2002 IEEE Symposium on Security and Privacy*, pages 202–15, 2002.

[4] United States Department of Defense Data Fusion SubPanel of the Joint Directors of Laboratories. Technical panel for C3. In *Data Fusion Lexicon*, October 1991.

[5] Security Focus. Bugtraq vulnerability database. `http://www.securityfocus.org/bid`, 2006.

[6] A. Fuchsberger. Intrusion detection systems and intrusion prevention systems. *Information Security Technical Report*, 10(3):134–9, 2005.

[7] David L. Hall and James Llinas. An introduction to multisensor data fusion. In *Proceedings of the IEEE*, volume 85, pages 6–23, January 1997.

[8] Jared Holsopple, Shanchieh Jay Yang, and Moises Sudit. TANDI: Threat assessment for networked data and information. In *Proceedings of SPIE, Defense and Security Symposium*, volume 6242, April 2006.

[9] J. Howard and T. Longstaff. A common language for computer security incidents, 1998.

[10] Tenable Network Security Inc. Nessus vulnerability scanner. `http://www.nessus.org/`, 2007.

[11] Insecure.org. Nmap (Network Mapper): a free open source utility for network exploration or security auditing. http://insecure.org/nmap/, 2007.

[12] SANS Institute. SANS critical vulnerability analysis archive. http://www.sans.org/newsletters/cva/.

[13] JDOM. A Java API to manipulate XML. http://www.jdom.org/, 2004.

[14] Ulf Lindqvist and Erland Jonsson. How to systematically classify computer security intrusions. In *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pages 154–163, Oakland, CA, USA, 1997.

[15] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White. Revisions and extensions to the JDL data fusion model II. In *Proceedings of The 7th International Conference on Information Fusion*, pages 1218–1230, June 2004.

[16] F. Massicotte, M. Couture, L. Briand, and Y. Labiche. Context-based intrusion detection using snort, nessus and bugtraq databases. In *Proceedings of the Third Annual Conference on Privacy, Security and Trust*, Fredericton, October 2005.

[17] Jelena Mirkovic and Peter Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. *Computer Communication Review*, 34(2):39–53, 2004.

[18] Mitre. Common vulnerabilities and exposures (CVE dictionary). http://cve.mitre.org, 2007.

[19] National Institute of Standards and Technology. National Vulnerability Database: a comprehensive cyber vulnerability resource. http://nvd.nist.gov/statistics.cfm, 2007.

[20] P. Ning, Y. Cui, D.S. Reeves, and D. Xu. Techniques and tools for analyzing intrusion alerts. *ACM Transactions on Information and Systems Security*, 7(2):274–318, 2004.

[21] Forum of Incident Response and Security Teams (FIRST). A complete guide to the common vulnerability scoring system. http://www.first.org/ cvss/cvss-guide.html, 2007.

[22] C. Phillips and L. P. Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 workshop for new security paradigms*, pages 71 – 79, New York, NY, USA, 1998.

[23] P.A. Porras, M.W. Fong, and A. Valdes. A mission-impact-based approach to infosec alarm correlation. In *Recent Advances in Intrusion Detection. 5th International Symposium, RAID 2002. Proceedings (Lecture Notes in Computer Science Vo.2516)*, pages 95 – 114, Zurich, Switzerland, 2002.

[24] Jean Roy, Stphane Paradis, and Mohamad Allouche. Threat evaluation for impact assessment in situation analysis systems. In *Proceedings of SPIE, Defense and Security Symposium, Signal Processing, Sensor Fusion, and Target Recognition XI Conference*, volume 4729, 2002.

[25] Sourcefire. Snort: an open source network intrusion prevention and detection system. `http://www.snort.org`, 2007.

[26] Moises Sudit, Adam Stotz, and Michael Holender. Situational awareness of a coordinated cyber attack. In *Proceedings of SPIE, Defense and Security Symposium*, pages 114–129, March 2005.

[27] Moises Sudit, Adam Stotz, Michael Holender, William Tagliaferri, and Kathie Canarelli. Measuring situation awareness and resolving inherent high-level fusion obstacles. In *Proceedings of SPIE, Defense and Security Symposium*, volume 6242, April 2006.

[28] United States Computer Emergency Readiness Team (US-CERT). US-CERT vulnerability note field. `http://www.kb.cert.org/vuls/html/fieldhelp`, 2006.

[29] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer. A comprehensive approach to intrusion detection alert correlation. *IEEE Transactions on Dependable and Secure Computing*, 1(3):146 – 169, July-Sep 2004.

[30] S. Vidalis and A. Jones. Using vulnerability trees for decision making in threat assessment. Technical report, University of Glamorgan, School of Computing, Wales, UK, June 2003.

[31] G. Vigna, F. Valeur, J. Zhou, and R.A. Kemmerer. Composable tools for network discovery and security analysis. In *Proceedings of 18th Annual Computer Security Applications Conference*, pages 14–24, Las Vegas, NV, USA, 2002.